

A MATLAB Script for Interplanetary Gravity-Assist Trajectory Design and Optimization

This document describes a MATLAB script called `flyby_snopt` that can be used to design and optimize interplanetary trajectories that include a single gravity assist maneuver. The user specifies the launch, flyby and destination planets, and the desired flyby altitude. The algorithm also requires initial guesses for the launch, flyby and arrival calendar dates. This script searches for a *patched-conic* gravity-assist trajectory that satisfies the flyby mission constraints (V_∞ matching and user-defined flyby altitude) and minimizes the launch, arrival or total *impulsive* delta-v for the mission. The type of delta-v optimization is specified by the user.

The planet positions and velocities are modeled using the JPL DE 421 ephemeris. The trajectory optimization is performed with the SNOPT nonlinear programming (NLP) algorithm.

The `flyby_snopt` software also provides a graphic display of the planet orbits and the interplanetary transfer trajectory.

Typical user interaction

The following is typical user interaction with this MATLAB application. This example is an Earth-Venus gravity-assist-Mars mission with a flyby altitude at Venus of 500 kilometers. The user inputs for this example are in bold font.

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

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

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:

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

For fixed launch, flyby or arrival times, the lower and upper bounds should be set to 1.0d-4.

```
program flyby_snopt
gravity-assist trajectory analysis

departure - calendar date guess

please input the calendar date
(1 <= month <= 12, 1 <= day <= 31, year = all digits!)
? 2,1,2009
```

please input the departure date search boundary in days
? 30

flyby - calendar date guess

please input the calendar date
(1 <= month <= 12, 1 <= day <= 31, year = all digits!)
? 6,1,2009

please input the flyby date search boundary in days
? 30

arrival - calendar date guess

please input the calendar date
(1 <= month <= 12, 1 <= day <= 31, year = all digits!)
? 2,1,2010

please input the arrival date search boundary in days
? 30

planet menu

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

please select the departure planet
? 3

planet menu

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

please select the flyby planet
? 2

planet menu

- <1> Mercury

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

```
please select the arrival planet
? 4
```

```
please input the flyby altitude (kilometers)
? 500
```

```
optimization menu
```

```
<1> minimize departure delta-v
<2> minimize arrival delta-v
<3> minimize total delta-v
```

```
selection (1, 2 or 3)
? 3
```

Optimal solution and trajectory graphics

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

```
program flyby_snopt
```

```
gravity assist trajectory analysis
```

```
LAUNCH CONDITIONS
=====
```

```
departure TDB calendar date    26-Jan-2009
```

```
departure TDB time              22:46:59.800
```

```
departure TDB Julian Date      2454858.449303
```

```
heliocentric launch delta-v vector and magnitude
(Earth mean ecliptic and equinox of J2000)
```

```
-----
launch delta-vx                 3598.330984 meters/second
```

```
launch delta-vy                 -107.513676 meters/second
```

```
launch delta-vz                 -4086.546917 meters/second
```

```
launch delta-v                  5446.045425 meters/second
```

```
launch hyperbola characteristics
(Earth mean equator and equinox of J2000)
```

asymptote right ascension 22.993112 degrees
asymptote declination -44.131177 degrees
launch energy 29.659411 (km/sec)^2

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

 sma (km) eccentricity inclination (deg) argper (deg)
1.2915349679e+008 1.5837709388e-001 8.4699120851e+000 3.3651002963e+002

 raan (deg) true anomaly (deg) arglat (deg) period (days)
3.0699636094e+002 2.0349519046e+002 1.8000522009e+002 2.9300131023e+002

 rx (km) ry (km) rz (km) rmag (km)
-8.86566261862594e+007 +1.17644805517813e+008 -1.97679458855093e+003 +1.47310208865380e+008

 vx (kps) vy (kps) vz (kps) vmag (kps)
-2.06860993612314e+001 -1.81557972184610e+001 -4.08724078800003e+000 +2.78254059571411e+001

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

 sma (km) eccentricity inclination (deg) argper (deg)
1.4973168636e+008 1.7489829256e-002 1.5179786566e-003 1.8766357058e+002

 raan (deg) true anomaly (deg) arglat (deg) period (days)
2.7657006597e+002 2.2767887557e+001 2.1043145813e+002 3.6574709272e+002

 rx (km) ry (km) rz (km) rmag (km)
-8.86566261862594e+007 +1.17644805517813e+008 -1.97679458855093e+003 +1.47310208865380e+008

 vx (kps) vy (kps) vz (kps) vmag (kps)
-2.42844303454548e+001 -1.80482835421561e+001 -6.93870616159664e-004 +3.02568024831238e+001

FLYBY CONDITIONS
=====

flyby TDB calendar date 02-Jun-2009
flyby TDB time 08:17:49.200
flyby TDB Julian Date 2454984.845708
launch-to-flyby time 126.396405 days

v-infinity in 7461.895071 meters/second
v-infinity out 7461.895266 meters/second

flyby altitude 499.999868 kilometers

maximum turn angle 58.793028 degrees
actual turn angle 56.203197 degrees

heliocentric delta-v 7029.649832 meters/second
max heliocentric delta-v 7326.580266 meters/second

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

 sma (km) eccentricity inclination (deg) argper (deg)
-5.8343985365e+003 2.1229777717e+000 7.2580123818e+001 3.8961669680e+001

```

      raan (deg)      true anomaly (deg)  arglat (deg)
2.3728101950e+002  0.0000000000e+000  3.8961669680e+001

      rx (km)          ry (km)          rz (km)          rmag (km)
-1.71602837531439e+003  -4.95285439259147e+003  +3.93088690478186e+003  +6.55189986776983e+003

      vx (kps)          vy (kps)          vz (kps)          vmag (kps)
+6.66637906058628e+000  +5.01736718015269e+000  +9.23200936203104e+000  +1.24436562175482e+001

```

b-plane coordinates of incoming hyperbola
(Earth mean ecliptic and equinox of J2000)

```

-----
b-magnitude          10926.123827  kilometers
b dot r              -5624.821633
b dot t              9367.046678
b-plane angle        329.015588  degrees
v-infinity           7461.895071  meters/second
r-periapsis          6551.899868  kilometers
decl-asymptote       69.561604  degrees
rasc-asymptote       359.934065  degrees

```

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

```

-----
      sma (km)      eccentricity      inclination (deg)      argper (deg)
1.0820914211e+008  6.7937075119e-003  3.3945276791e+000  5.4791401614e+001

      raan (deg)      true anomaly (deg)      arglat (deg)      period (days)
7.6652371284e+001  1.6186109023e+002  2.1665249184e+002  2.2470163874e+002

      rx (km)          ry (km)          rz (km)          rmag (km)
+4.29752824588006e+007  -9.99954916183663e+007  -3.84950418649687e+006  +1.08907262975865e+008

      vx (kps)          vy (kps)          vz (kps)          vmag (kps)
+3.19380484864012e+001  +1.370914444690867e+001  -1.65551062176933e+000  +3.47954062860716e+001

```

ARRIVAL CONDITIONS
=====

```

arrival TDB calendar date      14-Jan-2010
arrival TDB time                03:40:25.797
arrival TDB Julian Date        2455210.653076
flyby-to-arrival time          225.807368 days

```

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

```

-----
arrival delta-vx              3467.201759 meters/second
arrival delta-vy              2289.215414 meters/second
arrival delta-vz              -755.119378 meters/second

arrival delta-v                4222.819026 meters/second

```

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

```

-----
      sma (km)      eccentricity      inclination (deg)      argper (deg)
1.7688276799e+008  3.9074776462e-001  2.4686070296e+000  3.2054723491e+002

      raan (deg)      true anomaly (deg)      arglat (deg)      period (days)
3.4838070964e+002  1.7366712574e+002  1.3421436065e+002  4.6961136376e+002

```

rx (km)	ry (km)	rz (km)	rmag (km)
-1.32035624490053e+008	+2.06286504624848e+008	+7.56469827644770e+006	+2.45040308489908e+008
vx (kps)	vy (kps)	vz (kps)	vmag (kps)
-1.60226322091160e+001	-8.71332573270333e+000	-5.07076868727946e-001	+1.82456546932027e+001

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

```
-----
```

sma (km)	eccentricity	inclination (deg)	argper (deg)
2.2794303068e+008	9.3339698744e-002	1.8489037794e+000	2.8655658337e+002
raan (deg)	true anomaly (deg)	arglat (deg)	period (days)
4.9525421522e+001	1.4654795299e+002	7.3104536361e+001	6.8698900475e+002

rx (km)	ry (km)	rz (km)	rmag (km)
-1.32035624490053e+008	+2.06286504624848e+008	+7.56469827644770e+006	+2.45040308489908e+008
vx (kps)	vy (kps)	vz (kps)	vmag (kps)
-1.94898339683834e+001	-1.10025411468527e+001	+2.48042509322756e-001	+2.23823829135722e+001

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

```
-----
```

sma (km)	eccentricity	inclination (deg)	argper (deg)
2.2794303068e+008	9.3339698744e-002	1.8489037794e+000	2.8655658337e+002
raan (deg)	true anomaly (deg)	arglat (deg)	period (days)
4.9525421522e+001	1.4654795299e+002	7.3104536361e+001	6.8698900475e+002

rx (km)	ry (km)	rz (km)	rmag (km)
-1.32035624490001e+008	+2.06286504624876e+008	+7.56469827644935e+006	+2.45040308489904e+008
vx (kps)	vy (kps)	vz (kps)	vmag (kps)
-1.94898339683872e+001	-1.10025411468467e+001	+2.48042509322977e-001	+2.23823829135726e+001

MISSION SUMMARY
=====

total delta-v	9668.864451 meters/second
total energy	47.491611 (km/sec)^2
total mission duration	352.203773 days

SOI ENTRANCE CONDITIONS
=====

TDB calendar date	01-Jun-2009
TDB time	09:20:50.417
TDB Julian Date	2454983.889472

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

```
-----
```

sma (km)	eccentricity	inclination (deg)	argper (deg)
-5.9511194446e+003	1.0019810339e+000	1.0559757015e+002	2.5301764987e+002
raan (deg)	true anomaly (deg)	arglat (deg)	
4.8426708131e+001	1.8363818418e+002	7.6655834048e+001	

rx (km)	ry (km)	rz (km)	rmag (km)
+2.15002548438914e+005	-5.80406211733818e+002	+5.77556116222914e+005	+6.16277129295818e+005

```

          vx (kps)          vy (kps)          vz (kps)          vmag (kps)
-2.60506298761315e+000  +3.56958434068488e-003  -6.98968590934492e+000  +7.45936156937437e+000

```

```

CLOSEST APPROACH CONDITIONS
=====

```

```

TDB calendar date      02-Jun-2009
TDB time                07:32:25.677
TDB Julian Date        2454984.814186

```

```

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

```

```

          sma (km)          eccentricity          inclination (deg)          argper (deg)
-5.9498809746e+003  1.0022331397e+000  1.0680922944e+002  2.5437511796e+002

          raan (deg)          true anomaly (deg)          arglat (deg)
5.4097013129e+001  8.8716763841e-007  2.5437511885e+002

          rx (km)          ry (km)          rz (km)          rmag (km)
-5.09594809961748e+000  -7.28797097893550e-001  -1.22491751575514e+001  +1.32869155293570e+001

          vx (kps)          vy (kps)          vz (kps)          vmag (kps)
+1.10993117621459e+002  +1.82701652104185e+002  -5.70460961203619e+001  +2.21254656275304e+002

```

```

b-plane coordinates at closest approach
(Earth mean ecliptic and equinox of J2000)
-----

```

```

b-magnitude          397.853835 kilometers
b dot r              -223.016638
b dot t              -329.471172
b-plane angle        214.093785 degrees
v-infinity           7389.125529 meters/second
r-periapsis          13.286916 kilometers
decl-asymptote       -69.561158 degrees
rasc-asymptote       179.940197 degrees

flight path angle    0.000000 degrees

```

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

```

would you like to plot this trajectory (y = yes, n = no)
?

```

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

```

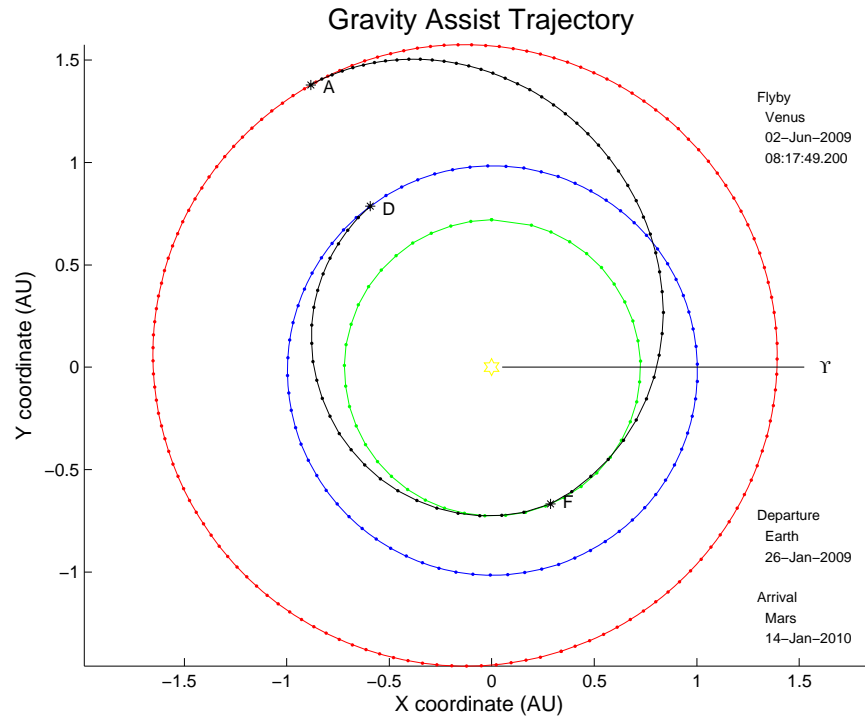
please input the plot step size (days)
?

```

A plot step size between 5 and 10 days is recommended.

The following is the 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 launch planet is labeled with an L, the flyby planet is labeled with an F and the arrival planet is labeled with an A. The location of the launch,

flyby and arrival planets at the launch time is marked with an asterisk. Please note that the axes are labeled in Astronomical Units.



Technical discussion

The computational steps used to create an initial guess for the spacecraft state, and the launch 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 launch delta-v vector from the launch 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) estimate 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

Lambert's Problem

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

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

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

From the following form of Kepler's equation

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

we can write

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

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

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

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

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

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

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

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

With the two substitutions given by

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

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

the time equation becomes

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

From the elliptic relationships given by

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

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

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

and some more manipulation, we have the following equations:

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

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

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

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

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

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

With the use of the half angle formulas given by

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

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

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

A discussion about the angles 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.

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

Gravity-assist flight mechanics

The vector relationships between the incoming v-infinity vector \mathbf{v}_∞^- , the outgoing v-infinity vector \mathbf{v}_∞^+ at the flyby planet, 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 hyperbolic trajectory during the flyby is determined from

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

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 gazes the planet’s surface. It is given by

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

where r_s is the radius of the flyby planet. 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 measured with respect to a spherical planet and is given by $h = r - r_p$.

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 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 two nonlinear equality *point constraints*:

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

$$h_{fb} - h_t = 0$$

The first equation is the v-infinity matching constraint and the second equation is the (positive) flyby altitude constraint. In the second expression h_{fb} is the actual flyby altitude and h_t is the user-defined or “targeted” flyby altitude relative to a spherical planet.

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

The right ascension of the asymptote is determined from

$$\alpha = \tan^{-1}(\Delta v_y, \Delta v_z)$$

and the geocentric declination of the asymptote is given by

$$\delta = 90^\circ - \cos^{-1}(\Delta v_z)$$

where $\Delta v_x, \Delta v_y, \Delta v_z$ are the components of the unit Δv vector.

In typical interplanetary “targeting specs”, the right ascension is called RLA and the declination is called DLA. The corresponding launch energy is called C3. C3 is equal to twice the specific (per unit mass) orbital energy of the departure trajectory.

In this script the heliocentric planetary coordinates and therefore the Δv vectors are computed in the J2000 ecliptic and equinox coordinate system. In order to determine the orientation of the departure hyperbola, the delta-v vector must be transformed to the Earth equatorial frame.

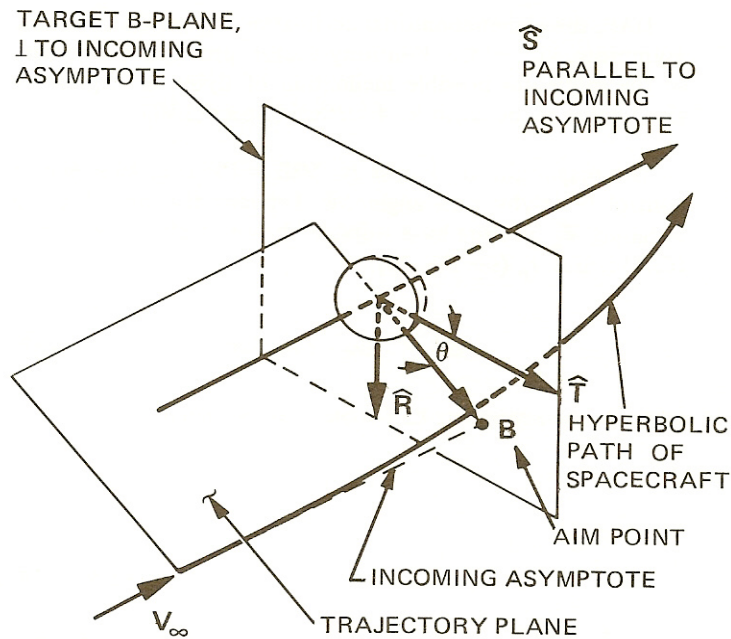
The required transformation is given by

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

where $\Delta \mathbf{V}_{ec}$ is the delta-velocity vector in the ecliptic frame, and $\Delta \mathbf{V}_{eq}$ is the delta-velocity vector in the Earth equatorial frame.

The B-plane

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



The 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 or outgoing hyperbola.

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

angular momentum vector

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

radius rate

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

semiparameter

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

semimajor axis

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

orbital eccentricity

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

true anomaly

$$\cos \theta = \frac{p - r}{er} \quad \sin \theta = \frac{\dot{r}h}{e\mu} \rightarrow \theta = \tan^{-1}(\sin \theta, \cos \theta)$$

B-plane magnitude

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

fundamental vectors

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

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

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

S vector

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

B vector

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

T vector

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

R vector

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

Calculating the Sphere-of-Influence Entrance and Closest Approach

This section describes the numerical methods used to predict the time at which the spacecraft enters the sphere-of-influence (SOI) of the flyby planet and the time of closest approach to the flyby planet. The search algorithm uses a numerical integration method embedded within a root-finding function. These calculations can be used to assess the difference between the patched-conic and numerically integrated solution for the flyby hyperbola.

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 source code from the main script that predicts SOI entrance. It uses the root-finder option of the built-in ODE45 function to evaluate the heliocentric equations of motion defined in a MATLAB function named `heqm` while searching for the time at which the flyby planet-to-spacecraft distance equals the SOI radius of the flyby planet.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% integrate trajectory from first impulse
% to SOI entrance at flyby planet
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set up for ode45

options = odeset('RelTol', 1.0e-6, 'AbsTol', 1.0e-8, 'Events', @soi_event);

% define maximum search time (seconds)

tof = 86400.0 * (jdate2 - jdate1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% solve for SOI entrance conditions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[t, ysol, tevent, yevent, ie] = ode45(@heqm, [0 tof], [ritol vitol], options);

```

The following is the MATLAB source code that computes the scalar value of the SOI objective function.

```

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

% entrance to SOI of flyby planet event function

% Orbital Mechanics with Matlab

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

global ip2 rsoi jdate1 rp2sc vp2sc

% current TDB julian date

```

```

jdate = jdate1 + t/ 86400.0d0;

% compute the position and velocity of flyby planet

[rp, vp] = p2000(ip2, jdate);

rsc = y(1:3);

vsc = y(4:6);

% vector from flyby planet to spacecraft

rp2sc = rp - rsc;

vp2sc = vp - vsc;

% objective function

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

isterminal = 1;

direction = [];

```

The software uses a similar technique to predict the spacecraft flight conditions at closest approach to the flyby planet. The flyby planet-centered initial conditions for the search are determined from the state vector of the spacecraft at SOI entrance.

For close approach prediction, the script uses the following objective function

$$\Delta = \frac{\mathbf{r}_{p-sc} \cdot \mathbf{v}_{p-sc}}{|\mathbf{r}_{p-sc} \cdot \mathbf{v}_{p-sc}|}$$

which is simply the sine of the flight path angle. Since the spacecraft's orbit within the SOI is hyperbolic, periapsis occurs whenever the flight path angle is zero.

During the search for closest approach conditions, the flyby planet-centered equations of motion subject to the point-mass gravity of the sun and the flyby planet are computed using

$$\mathbf{a} = \mathbf{a}_p + \mathbf{a}_s = -\mu_p \frac{\mathbf{r}_{p-sc}}{|\mathbf{r}_{p-sc}|^3} - \mu_s \left\{ \frac{\mathbf{r}_{s-sc}}{|\mathbf{r}_{s-sc}|^3} + \frac{\mathbf{r}_p}{|\mathbf{r}_p|^3} \right\}$$

In this equation, \mathbf{a} is the total acceleration on the spacecraft and \mathbf{r}_{s-sc} is the heliocentric position vector of the spacecraft given by

$$\mathbf{r}_{s-sc} = \mathbf{r}_{p-sc} + \mathbf{r}_p$$

where \mathbf{r}_p is the heliocentric position vector of the flyby planet.

SNOPT algorithm implementation

This section provides details about the part of the `flyby_snopt` MATLAB script that solves this nonlinear programming (NLP) problem using the SNOPT algorithm. In this classic trajectory optimization problem, the launch, flyby and arrival calendar dates are the *control variables* and the user-specified scalar ΔV is the *objective function* or *performance index*.

MATLAB versions of SNOPT for several computer platforms can be found at Professor Philip Gill's web site which is located at <http://scicomp.ucsd.edu/~peg/>. A PDF version of the SNOPT user's manual is also available at Professor Gill's website.

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

```
% control variables initial guess

xg(1) = jdatei1 - jdate0;

xg(2) = jdatei2 - jdate0;

xg(3) = jdatei3 - jdate0;

xg = xg';
```

where `jdatei1`, `jdatei2` and `jdatei3` are the initial user-provided launch, flyby and arrival calendar date guesses, and `jdate0` is a reference Julian date equal to 2451544.5 (January 1, 2000). This offset value is used to *scale* the control variables.

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

```
% bounds on control variables

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

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

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

xlwr = xlwr';
xupr = xupr';

xlwr = xlwr';
xupr = xupr';
```

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

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

```
% bounds on objective function

flow(1) = 0.0d0;
fupp(1) = +Inf;
```

The following MATLAB source code enforces the equality flyby constraints:

```
% load mission constraints

y(2) = fbalt - alttar;

y(3) = vinfm_in - vinfm_out;
```

In this code `fbalt`, `vinfm_in`, and `vinfm_out` are the current flyby altitude, incoming and outgoing v-infinity speeds, respectively. `alttar` is the user-defined flyby altitude.

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

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

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

The `flyby_snopt` script will also create summary files and set other algorithm options such as derivative and iteration limits. The MATLAB code that performs these functions is as follows:

```
% set SNOPT options

snprintf('flyby_snopt.out');

snsummary('flyby_snopt.sum');

snseti('derivative option', 0);

snseti('minor iterations limit', 1000);
```

Please consult the SNOPT user's guide for an explanation of these program options.

References and bibliography

“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.

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

“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.

“An Essay on the Application and Principle of Gravity-assist Trajectories for Space Flight”, Victor C. Clarke, Jr., JPL, April 10, 1970.

“The invention that opened the solar system to exploration”, Michael A. Minovitch, *Planetary and Space Science*, 58 (2010) 885-892.

“A Graphical Method for Gravity-assist Trajectory Design”, Nathan J. Strange and James M. Longuski, AIAA 2000-4030.

Lecture Notes, Professor Harm Bunning, The University of Michigan, Department of Aerospace Engineering.

APPENDIX A

Contents of the Simulation Summary

This appendix is a brief summary of the information contained in the simulation summary screen 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 (km) = semimajor axis in kilometers

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 (kps) = 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-asymptote = declination of incoming v-infinity vector in degrees
rasc-asymptote = right ascension of incoming v-infinity vector in degrees
fpa = flight path angle in degrees