

Program ca_sc2body

Closest Approach between a Spacecraft and Celestial Body

This document is the user's manual for a Windows compatible computer program called `ca_sc2body` which can be used to estimate the time of closest approach between an interplanetary spacecraft and a planet, comet or other celestial body. The heliocentric motion of the spacecraft is modeled using continuous, low-thrust propulsion. This information can be used to create an initial guess for the `ilt_socx` computer program and other types of trajectory analyses.

The software also includes the option to propagate a spacecraft's trajectory for a user-defined time duration. This allows the user to access the effects of such trajectory characteristics as launch date, launch C3, propulsion characteristics and so forth on the long-term orbital evolution of the spacecraft.

The important numerical methods used in this computer program are as follows:

- Equations of motion based on modified equinoctial orbital elements
- JPL DE421 planetary ephemeris
- Continuous, low-thrust tangential steering
- Brent's method for one-dimensional minimization
- Runge-Kutta-Fehlberg 7(8) numerical integrator

The `ca_sc2body` computer program was written and compiled with Intel Visual Fortran, version 11.1. The fundamental time argument used within the software is Barycentric Dynamical Time (TDB).

Program execution

An input file created by the user can be run from the command line or a simple batch file with a statement similar to the following:

```
ca_sc2body tempell.in
```

If the software is executed without an input file on the command line, the computer program will display the following information screen and file name prompt:

```
*****  
*           program ca_sc2body           *  
*                                         *  
*       closest approach between         *  
* a spacecraft and celestial body       *  
*                                         *  
*           January 27, 2011            *  
*****
```

```
please input the name of the simulation definition file
```

At this point the user should supply the name of a compatible input file. The next section of this document describes the proper format for an input file for this computer program.

Typical input file

The `ca_sc2body` software is “data-driven” by a user-created text file. The following is a typical input file used by this computer program. This example is an Earth-to-Mars trajectory with an initial launch C_3 of $4.625 \text{ km}^2/\text{sec}^2$.

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

In the following discussion the actual input file contents are in *courier* font and all explanations are in *times italic* font.

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

```
*****
** interplanetary trajectory transfer time
** patched-conic heliocentric motion
** continuous, low-thrust maneuver
** Earth-to-Mars - February 25, 2005
*****
```

The first program input is an integer that specifies the type of simulation.

```
simulation type (1 = propagation, 2 = close approach)
2
```

The next inputs are the initial launch energy, spacecraft mass, thrust magnitude and specific impulse.

```
C3L = launch specific orbital energy ([km/sec]**2)
4.625d0

initial spacecraft mass (kilograms)
1171.1

thrust magnitude (newtons)
0.16831

specific impulse (seconds)
3070.0

*****
* LAUNCH CONDITIONS *
*****
```

The next three inputs define the launch calendar date. Be sure to include all four digits of the calendar year.

```
launch calendar date (month, day, year)
7, 10, 2005
```

The next program input is an integer that specifies the launch planet.

```

*****
* launch planet *
*****
1 = Mercury
2 = Venus
3 = Earth
4 = Mars
5 = Jupiter
6 = Saturn
7 = Uranus
8 = Neptune
9 = Pluto
-----
3

```

```

*****
* ARRIVAL CONDITIONS *
*****

```

The next input is an integer that defines the arrival planet, comet or asteroid.

```

*****
* arrival celestial body *
*****
1 = Mercury
2 = Venus
3 = Earth
4 = Mars
5 = Jupiter
6 = Saturn
7 = Uranus
8 = Neptune
9 = Pluto
0 = asteroid/comet
-----
4

```

The next input is a number that represents either a guess for the transfer time to closest approach or the time duration for the orbit propagation option.

```

transfer time guess or propagation duration (days)
400.0d0

```

This section contains inputs for a user-defined asteroid or comet. This information includes such things as the TDB calendar date of perihelion passage and the object's heliocentric orbital elements. These orbital elements must be specified relative to the Earth mean ecliptic and equinox of J2000.

```

*****
* asteroid/comet orbital elements *
* (heliocentric, ecliptic J2000) *
*****

asteroid/comet name
Tempel 1

calendar date of perihelion passage (month, day, year)
7, 5.3153, 2005

perihelion distance (au)
1.506167

orbital eccentricity (nd)
0.517491

```

```
orbital inclination (degrees)
10.5301

argument of perihelion (degrees)
178.8390

longitude of the ascending node (degrees)
68.9734
```

The final input is the name of the data file that will contain the trajectory information.

```
name of solution output file
e2m1.csv
```

Program solution and graphics

For close approach simulations, the software will provide a screen display of close approach conditions similar to the following:

```
program ca_sc2body

closest approach between a spacecraft and celestial body

(continuous, low-thrust tangential steering)

input data file ==> e2m1.in

close approach conditions - DE421 ephemeris
-----

calendar date          03/02/2006

TDB time                20:58:31.434

TDB Julian date        2453797.37397494

spacecraft-to-body distance  8.076857954393460E-002  AU

spacecraft-to-body distance  12082807.5185093          kilometers

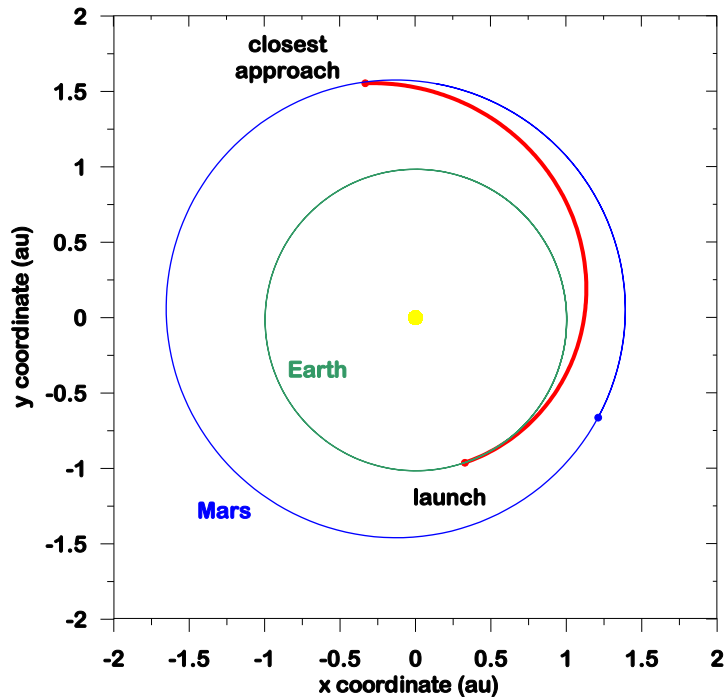
transfer time           235.873974935370          days
```

For simple orbit propagation simulations, the software does not produce any screen output. However, it does create the two data files explained in the next section.

The `ca_sc2body` software will also create two comma-separated-variable (csv) output files. The first file contains the heliocentric, ecliptic state vector of the spacecraft, and the second file (`planets.csv`) contains the state vectors of both celestial bodies. Please see Appendix A for additional information about the contents of these data files. The following plot is a view of the trajectory and planetary orbits from the north pole of the ecliptic looking down on the ecliptic plane.

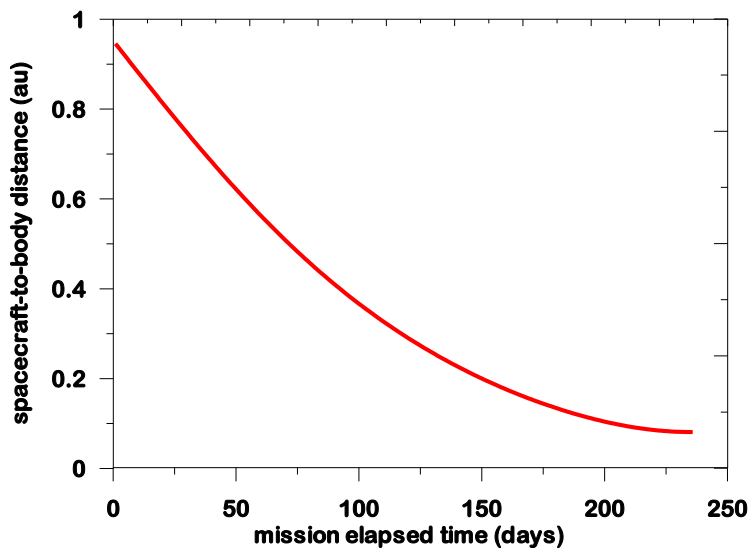
Continuous, Low-thrust Trajectory Analysis Earth-to-Mars Tangential Thrust

Mass = 1171.1 kg Thrust = 168.31 mN Isp = 3070 s



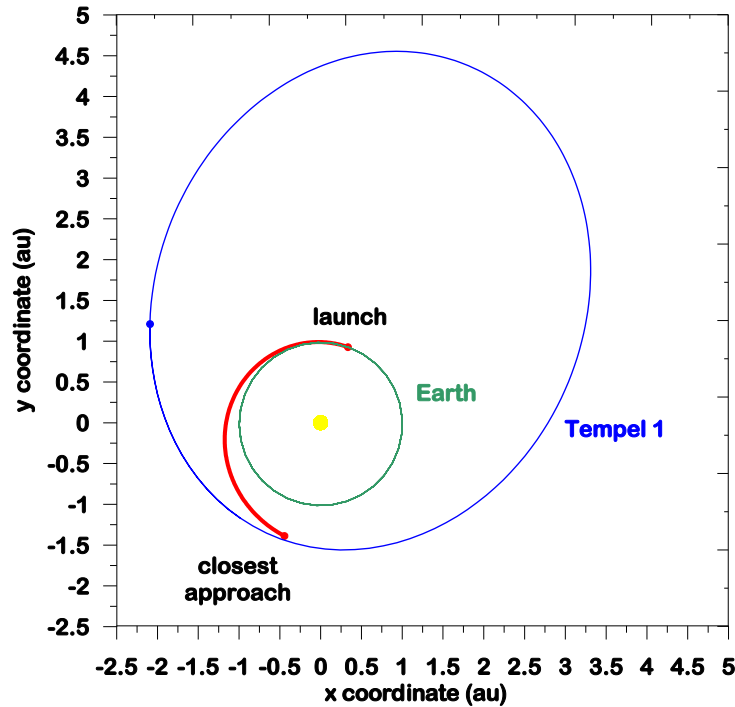
Continuous, Low-thrust Trajectory Analysis Earth-to-Mars Tangential Thrust

Mass = 1171.1 kg Thrust = 168.31 mN Isp = 3070 s



The following is the trajectory plot for the Tempel 1 example input file (`tempel1.in`) included with this computer program.

Continuous, Low-thrust Trajectory Analysis
Earth-to-Tempel1 Tangential Thrust
Mass = 545 kg Thrust = 92.3155 mN Isp = 3337 s



Technical Discussion

An initial guess for the transfer time can be created by performing a one-dimensional minimization while numerically integrating the equations of motion with tangential thrusting. This process determines future close approach conditions between the spacecraft in its *coplanar* transfer orbit and the destination celestial body. This computational process is performed using Brent's minimization algorithm with the following objective function

$$f(t) = \Delta r(t) = |\mathbf{r}_p - \mathbf{r}_{sc}|$$

where \mathbf{r}_p is the heliocentric position vector of the arrival planet and \mathbf{r}_{sc} is the heliocentric position vector of the spacecraft at any simulation time t . If the separation distance Δr is "close" enough (say 0.01 to perhaps 0.2 AU), the trajectory time provides a good initial guess. This close approach algorithm also uses the launch date, thrust and spacecraft mass provided by the user.

During the low-thrust interplanetary transfer to an *outer planet*, the algorithm assumes *tangential thrusting* such that the unit thrust vector in the modified equinoctial frame at all times is simply $\mathbf{u}_T = [0 \ 1 \ 0]^T$. Please note that this approach creates a *coplanar* solution. This initial guess algorithm uses the launch date, thrust and spacecraft mass provided by the user. For orbital transfer to an *inner planet*, the unit thrust vector is $\mathbf{u}_T = [0 \ -1 \ 0]^T$.

To model an impulsive delta-v at launch, the initial guess algorithm assumes that this maneuver is applied along the direction of the velocity vector of the departure planet. For outer planet missions, this direction is along the direction of orbital motion. For inner planet missions, this vector is aligned opposite to the direction of orbital motion.

The unit thrust vector of this maneuver in the heliocentric inertial coordinate system is

$$\mathbf{u}_T = \frac{\mathbf{r}_p}{|\mathbf{r}_p|}$$

where \mathbf{r}_p is the heliocentric position vector of the launch planet at the launch calendar date.

Modified equinoctial orbital elements

The modified equinoctial orbital elements are a set of orbital elements that are useful for trajectory analysis and optimization. They are valid for circular, elliptic, and hyperbolic orbits. These equations exhibit no singularity for zero eccentricity and orbital inclinations equal to 0 and 90 degrees. However, two components of the orbital element set are singular for an orbital inclination of 180 degrees.

The relationship between direct modified equinoctial and classical orbital elements is defined by the following definitions

$$p = a(1 - e^2)$$

$$f = e \cos(\omega + \Omega)$$

$$g = e \sin(\omega + \Omega)$$

$$h = \tan(i/2) \cos \Omega$$

$$k = \tan(i/2) \sin \Omega$$

$$L = \Omega + \omega + \theta$$

where

p = semiparameter

a = semimajor axis

e = orbital eccentricity

i = orbital inclination

ω = argument of periapsis

Ω = right ascension of the ascending node

θ = true anomaly

L = true longitude

The relationship between classical and modified equinoctial orbital elements is summarized as follows:

semimajor axis

$$a = \frac{p}{1 - f^2 - g^2}$$

orbital eccentricity

$$e = \sqrt{f^2 + g^2}$$

orbital inclination

$$i = 2 \tan^{-1} \left(\sqrt{h^2 + k^2} \right)$$

argument of periapsis

$$\omega = \tan^{-1} (g/f) - \tan^{-1} (k/h)$$

right ascension of the ascending node

$$\Omega = \tan^{-1} (k/h)$$

true anomaly

$$\theta = L - (\Omega + \omega) = L - \tan^{-1} (g/f)$$

The mathematical relationships between an inertial state vector and the corresponding modified equinoctial elements are summarized as follows:

position vector

$$\mathbf{r} = \begin{bmatrix} \frac{r}{s^2} (\cos L + \alpha^2 \cos L + 2hk \sin L) \\ \frac{r}{s^2} (\sin L - \alpha^2 \sin L + 2hk \cos L) \\ \frac{2r}{s^2} (h \sin L - k \cos L) \end{bmatrix}$$

velocity vector

$$\mathbf{v} = \begin{bmatrix} -\frac{1}{s^2} \sqrt{\frac{\mu}{p}} (\sin L + \alpha^2 \sin L - 2hk \cos L + g - 2f h k + \alpha^2 g) \\ -\frac{1}{s^2} \sqrt{\frac{\mu}{p}} (-\cos L + \alpha^2 \cos L + 2hk \sin L - f + 2g h k + \alpha^2 f) \\ \frac{2}{s^2} \sqrt{\frac{\mu}{p}} (h \cos L + k \sin L + f h + g k) \end{bmatrix}$$

where

$$\alpha^2 = h^2 - k^2$$

$$s^2 = 1 + h^2 + k^2$$

$$r = \frac{p}{w}$$

$$w = 1 + f \cos L + g \sin L$$

The system of first-order modified equinoctial equations of orbital motion are given by

$$\dot{p} = \frac{dp}{dt} = \frac{2p}{w} \sqrt{\frac{p}{\mu}} \Delta_t$$

$$\dot{f} = \frac{df}{dt} = \sqrt{\frac{p}{\mu}} \left[\Delta_r \sin L + [(w+1) \cos L + f] \frac{\Delta_t}{w} - (h \sin L - k \cos L) \frac{g \Delta_n}{w} \right]$$

$$\dot{g} = \frac{dg}{dt} = \sqrt{\frac{p}{\mu}} \left[-\Delta_r \cos L + [(w+1) \sin L + g] \frac{\Delta_t}{w} + (h \sin L - k \cos L) \frac{g \Delta_n}{w} \right]$$

$$\dot{h} = \frac{dh}{dt} = \sqrt{\frac{p}{\mu}} \frac{s^2 \Delta_n}{2w} \cos L$$

$$\dot{k} = \frac{dk}{dt} = \sqrt{\frac{p}{\mu}} \frac{s^2 \Delta_n}{2w} \sin L$$

$$\dot{L} = \frac{dL}{dt} = \sqrt{\mu p} \left(\frac{w}{p} \right)^2 + \frac{1}{w} \sqrt{\frac{p}{\mu}} (h \sin L - k \cos L) \Delta_n$$

where $\Delta_r, \Delta_t, \Delta_n$ are *non-two-body* perturbations in the radial, tangential and normal directions, respectively. For an interplanetary spacecraft, the radial direction is along the heliocentric radius vector of the spacecraft measured positive in a direction away from the gravitational center, the tangential direction is perpendicular to this radius vector measured positive in the direction of orbital motion, and the normal direction is positive along the angular momentum vector of the spacecraft's orbit.

The equations of orbital motion can also be expressed in vector form as follows:

$$\dot{\mathbf{y}} = \frac{d\mathbf{y}}{dt} = \mathbf{A}(\mathbf{y})\mathbf{P} + \mathbf{b}$$

where

$$\mathbf{A} = \begin{pmatrix} 0 & \frac{2p}{w} \sqrt{\frac{p}{\mu}} & 0 \\ \sqrt{\frac{p}{\mu}} \sin L & \sqrt{\frac{p}{\mu}} \frac{1}{w} [(w+1) \cos L + f] & -\sqrt{\frac{p}{\mu}} \frac{g}{w} [h \sin L - k \cos L] \\ -\sqrt{\frac{p}{\mu}} \cos L & \sqrt{\frac{p}{\mu}} [(w+1) \sin L + g] & \sqrt{\frac{p}{\mu}} \frac{f}{w} [h \sin L - k \cos L] \\ 0 & 0 & \sqrt{\frac{p}{\mu}} \frac{s^2 \cos L}{2w} \\ 0 & 0 & \sqrt{\frac{p}{\mu}} \frac{s^2 \sin L}{2w} \\ 0 & 0 & \sqrt{\frac{p}{\mu}} [h \sin L - k \cos L] \end{pmatrix}$$

and

$$\mathbf{b} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \sqrt{\mu p} \left(\frac{w}{p}\right)^2 \end{bmatrix}^T$$

The total non-two-body acceleration vector is given by

$$\mathbf{P} = \Delta_r \hat{\mathbf{i}}_r + \Delta_t \hat{\mathbf{i}}_t + \Delta_n \hat{\mathbf{i}}_n$$

where $\hat{\mathbf{i}}_r$, $\hat{\mathbf{i}}_t$ and $\hat{\mathbf{i}}_n$ are unit vectors in the radial, tangential and normal directions. These unit vectors can be computed from the inertial position vector \mathbf{r} and velocity vector \mathbf{v} according to

$$\hat{\mathbf{i}}_r = \frac{\mathbf{r}}{|\mathbf{r}|} \quad \hat{\mathbf{i}}_n = \frac{\mathbf{r} \times \mathbf{v}}{|\mathbf{r} \times \mathbf{v}|} \quad \hat{\mathbf{i}}_t = \hat{\mathbf{i}}_n \times \hat{\mathbf{i}}_r = \frac{(\mathbf{r} \times \mathbf{v}) \times \mathbf{r}}{|\mathbf{r} \times \mathbf{v}| |\mathbf{r}|}$$

For *unperturbed* two-body motion, $\mathbf{P} = 0$ and the first five equations of motion are simply $\dot{p} = \dot{f} = \dot{g} = \dot{h} = \dot{k} = 0$. Therefore, for two-body motion these modified equinoctial orbital elements are constant. The true longitude is often called the *fast variable* of this orbital element set.

The acceleration due to propulsive thrust can be expressed as

$$\mathbf{a}_T = \frac{T}{m} \hat{\mathbf{u}}$$

where T is the thrust, m is the spacecraft mass and $\hat{\mathbf{u}} = [u_r \quad u_t \quad u_n]$ is the unit pointing thrust vector expressed in the spacecraft-centered radial-tangential-normal coordinate system. The components of the

unit thrust vector can also be defined in terms of the in-plane pitch angle θ and the out-of-plane yaw angle ψ as follows:

$$\begin{aligned}u_r &= \sin \theta \\u_t &= \cos \theta \cos \psi \\u_n &= \cos \theta \sin \psi\end{aligned}$$

Finally, the pitch and yaw angles can be determined from the components of the unit thrust vector according to

$$\begin{aligned}\theta &= \sin^{-1}(u_r) \\ \psi &= \tan^{-1}(u_n, u_t)\end{aligned}$$

The pitch angle is positive above the “local horizontal” and the yaw angle is positive in the direction of the angular momentum vector.

Planetary ephemeris

The software models the planetary coordinates using the DE421 model from JPL. This algorithm provides position and velocity vectors of a planet relative to the Earth mean equator and equinox of J2000 (EME2000). Equatorial state vectors (position and velocity vectors) must be transformed to the ecliptic frame for use within the `ca_sc2body` computer program. The required transformation is

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

where \mathbf{r}_{ec} is the position vector in the ecliptic frame and \mathbf{r}_{eq} is the position vector in the equatorial frame. The same transformation is applied to the velocity vector.

Comet and asteroid ephemeris

The orbital elements of an asteroid or comet relative to the ecliptic and equinox of J2000 coordinate system must be provided by the user. These elements can be obtained from the ASTCOM database at JPL (<http://ssd.jpl.nasa.gov>), the MPC database at Harvard (<http://cfa-www.harvard.edu>) or the Bureau of Longitudes in Paris (<http://www.bdl.fr>).

These orbital elements consist of the following items:

- TDB calendar date of perihelion passage
- perihelion distance (AU)
- orbital eccentricity (non-dimensional)
- orbital inclination (degrees)
- argument of perihelion (degrees)
- longitude of ascending node (degrees)

The software determines the mean anomaly of the asteroid or comet at any simulation time using the following equation:

$$M = \sqrt{\frac{\mu_s}{a^3}} t_{pp} = \sqrt{\frac{\mu_s}{a^3}} (JD - JD_{pp})$$

where μ_s is the gravitational constant of the sun, a is the semimajor axis of the celestial body, and t_{pp} is the time since perihelion passage.

The semimajor axis is determined from the perihelion distance r_p and orbital eccentricity e according to

$$a = \frac{r_p}{(1 - e)}$$

Kepler's equation

This solution of Kepler's equation in this computer program is based on a numerical solution devised by Professor J.M.A. Danby at North Carolina State University. Additional information about this algorithm can be found in "The Solution of Kepler's Equation", *Celestial Mechanics*, **31** (1983) 95-107, 317-328 and **40** (1987) 303-312.

The initial guess for Danby's method is

$$E_0 = M + 0.85 \text{sign}(\sin M)e$$

The fundamental equation we want to solve is

$$f(E) = E - e \sin E - M = 0$$

which has the first three derivatives given by

$$f'(E) = 1 - e \cos E$$

$$f''(E) = e \sin E$$

$$f'''(E) = e \cos E$$

The iteration for an updated eccentric anomaly based on a current value E_n is given by:

$$\Delta(E_n) = -\frac{f}{f'}$$

$$\Delta^*(E_n) = -\frac{f}{f' + \frac{1}{2}\Delta f''}$$

$$\Delta_n(E_n) = -\frac{f}{f' + \frac{1}{2}\Delta f'' + \frac{1}{6}\Delta^2 f'''}$$

$$E_{n+1} = E_n + \Delta_n$$

This algorithm provides quartic convergence of Kepler's equation. This process is repeated until the following convergence test involving the fundamental equation is satisfied:

$$|f(E)| \leq \varepsilon$$

where ε is the convergence tolerance. This tolerance is hardwired in the software to $\varepsilon = 1.0e-10$.

Finally, the true anomaly can be calculated with the following two equations

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

$$\cos \theta = \cos E - e$$

and the four quadrant inverse tangent given by

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

If the orbit is hyperbolic, the initial guess is

$$H_0 = \log\left(\frac{2M}{e} + 1.8\right)$$

where H_0 is the hyperbolic anomaly. The fundamental equation and first three derivatives for this case are as follows:

$$f(H) = e \sinh H - H - M$$

$$f'(H) = e \cosh H - 1$$

$$f''(H) = e \sinh H$$

$$f'''(H) = e \cosh H$$

Otherwise, the iteration loop which calculates Δ, Δ^* , and so forth is the same. The true anomaly for hyperbolic orbits is determined with this next set of equations:

$$\sin \theta = \sqrt{e^2 - 1} \sinh H$$

$$\cos \theta = e - \cosh H$$

The true anomaly is then determined from a four quadrant inverse tangent evaluation of these two equations.

References and Bibliography

“A Set of Modified Equinoctial Orbital Elements”, M. J. H. Walker, B. Ireland and J. Owens, *Celestial Mechanics*, Vol. 36, pp. 409-419, 1985.

“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.

“Using Sparse Nonlinear Programming to Compute Low Thrust Orbit Transfers”, John T. Betts, *The Journal of the Astronautical Sciences*, Vol. 41, No. 3, July-September 1993, pp. 349-371.

“Equinoctial Orbit Elements: Application to Optimal Transfer Problems”, Jean A. Kechichian, AIAA 90-2976, AIAA/AAS Astrodynamics Conference, Portland, OR, 20-22 August 1990.

“Optimal Low Thrust Trajectories to the Moon”, John T. Betts and Sven O. Erb, *SIAM Journal on Applied Dynamical Systems*, Vol. 2, No. 2, pp. 144-170, 2003.

“Modern Astrodynamics”, Victor R. Bond and Mark C. Allman, Princeton Univeristy Press, 1996.

Appendix A

Contents of the CSV Files

This appendix is a brief summary of the information contained in the CSV data files produced by the `ca_sc2body` software. All output is computed and displayed in a heliocentric, Earth mean ecliptic and equinox of J2000 coordinate system.

The user-specified comma-separated-variable disk file contains the following information:

time (days) = simulation time since launch in days

rsc-x (au) = x-component of spacecraft's heliocentric position vector in astronomical units

rsc-y (au) = y-component of spacecraft's heliocentric position vector in astronomical units

rsc-z (au) = z-component of spacecraft's heliocentric position vector in astronomical units

rs2b (au) = distance from the spacecraft to the celestial body in astronomical units

The `planets.csv` data file contains the following information:

time (days) = simulation time since launch in days

rp1-x (au) = x-component of the launch planet position vector in astronomical units

rp1-y (au) = y-component of the launch planet position vector in astronomical units

rp1-z (au) = z-component of the launch planet position vector in astronomical units

rp2-x (au) = x-component of the destination body position vector in astronomical units

rp2-y (au) = y-component of the destination body position vector in astronomical units

rp2-z (au) = z-component of the destination body position vector in astronomical units

The value of the Astronomical Unit used in this computer program is 149597870.691 kilometers.