A MATLAB Script for Calculating Hyperbolic Coordinates

This brief memo is the user's manual for a MATLAB script named `demo_hyper` that can be used to calculate C_3 (twice the specific (per unit mass) orbital energy), RLA (the right ascension α_∞) and DLA (declination δ_∞) of the asymptote of a hyperbolic trajectory. The script can use both Cartesian state vector (position and velocity vectors) and classical orbital elements as the data source.

This computer program assumes that the hyperbolic targets, state vector and classical orbital elements are all in the same Earth-centered-inertial (ECI) coordinate system.

Input data file

The `demo_hyper` MATLAB script is "data-driven" by a simple text file created by the user. This section describes two typical input data files. 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. The annotation line also includes the correct units and when appropriate, the valid range of the input.

The following information illustrates the contents of a typical state vector (position and velocity vectors) input data file.

```
************************
* state vector data file
************************

eci position vector (kilometers)

-6.28143245744413e+003
-1.71886519445504e+003
-8.16419427413681e+002

eci velocity vector (kilometers/second)

+3.30316298967422e+000
-9.56155991173246e+000
-5.28351302498913e+000
```

Here are the contents of a typical classical orbital elements input data file.

```
****************************
* orbital elements data file
****************************

semimajor axis (kilometers)
(semimajor axis > 0)
-45361.7896303620

orbital eccentricity (non-dimensional)
(0 <= eccentricity < 1)
1.14468873590488
```

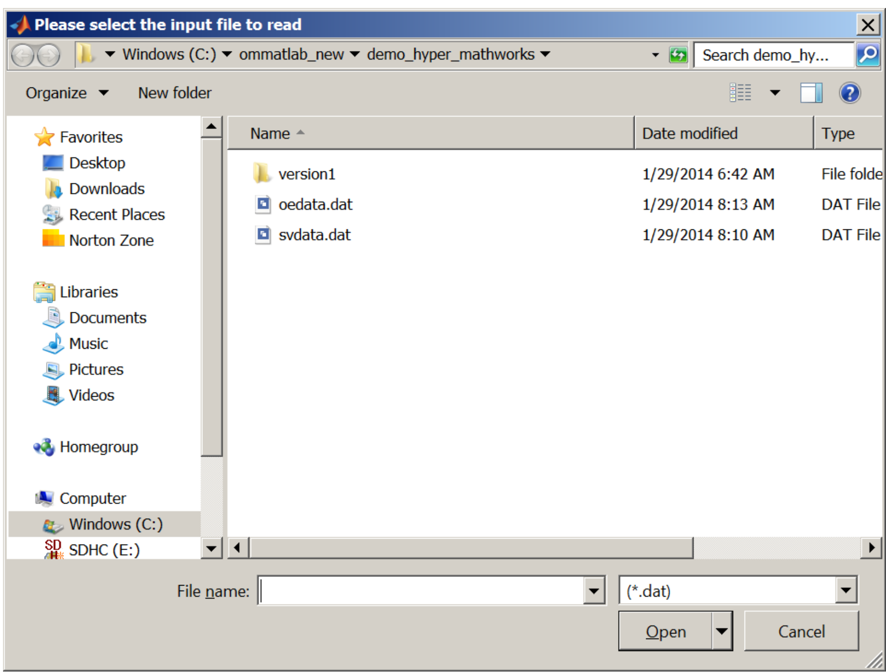
```
orbital inclination (degrees)
(0 <= inclination <= 180)
28.6442848562298

argument of perigee (degrees)
(0 <= argument of perigee <= 360)
195.039684255199

right ascension of the ascending node (degrees)
(0 <= RAAN <= 360)
2.03552732637654

true anomaly (degrees)
(0 <= true anomaly <= 360)
0.0
```

Script example

After typing `demo_hyper` at the MATLAB command line, the script will ask you for the type of coordinate data to use for the calculations with the following prompt;

```
please input the type of coordinates (1 = state vector, 2 = orbital elements)
?
```

The script will then ask you to select the name of a data file with the following screen display.



Be sure to select a data file that is compatible with the coordinate type you selected previously.

The file type defaults to names with a `*.dat` filename extension. However, you can select any `demo_hyper` 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.

The following is a typical user interaction with `demo_hyper` and calculation results.

```
A MATLAB Script for Computing Hyperbolic Coordinates
====================================================

please input the type of coordinates (1 = state vector, 2 = orbital elements)
? 2


rv2hyper function

specific orbital energy         8.787141  (km/sec)**2

asymptote right ascension     349.621260  degrees

asymptote declination          -6.697329  degrees


orb2hyper function

asymptote right ascension     349.621260  degrees

asymptote declination          -6.697329  degrees


state vector

        rx (km)                 ry (km)                 rz (km)                rmag (km)
 -6.28143245744429e+03  -1.71886519445511e+03  -8.16419427413715e+02  +6.56334000000017e+03

        vx (kps)                vy (kps)                vz (kps)               vmag (kps)
 +3.30316298967423e+00  -9.56155991173233e+00  -5.28351302498906e+00  +1.14127044808507e+01


classical orbital elements

        sma (km)              eccentricity        inclination (deg)         argper (deg)
 -4.53617896303620e+04  +1.14468873590488e+00  +2.86442848562298e+01  +1.95039684255199e+02

       raan (deg)          true anomaly (deg)         arglat (deg)
 +2.03552732637654e+00  +0.00000000000000e+00  +1.95039684255199e+02
```

Technical Discussion

This section provides additional details about the numerical algorithms used in this MATLAB script. The first part describes the algorithm used to convert a state vector to hyperbolic coordinates. The section part of this discussion describes the algorithm used to convert classical orbital elements to the corresponding hyperbolic coordinates.

Using position and velocity vectors

The asymptote unit vector of a hyperbolic orbit is given by

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

In this expression, α_∞ is the right ascension of the asymptote (RLA), and δ_∞ is the declination of the asymptote (DLA).

The asymptote unit vector at any trajectory time can be computed from

$$\hat{\mathbf{s}} = \frac{1}{1 + C_3 \dfrac{h^2}{\mu^2}} \left\{ \left(\frac{\sqrt{C_3}}{\mu} \right) \mathbf{h} \times \mathbf{e} - \mathbf{e} \right\} = \frac{1}{1 + C_3 \dfrac{p}{\mu}} \left\{ \left(\frac{\sqrt{C_3}}{\mu} \right) \mathbf{h} \times \mathbf{e} - \mathbf{e} \right\}$$

where **h** and **e** are the angular momentum and orbital eccentricity vectors, respectively. In the second expression, p is the semiparameter of the hyperbolic orbit which can be computed from

$$p = a\left(1 - e^2\right)$$

The angular momentum and eccentricity vectors are computed using the following equations;

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

$$\mathbf{e} = \frac{\mathbf{v} \times \mathbf{h}}{\mu} - \frac{\mathbf{r}}{r} = \frac{1}{\mu}\left[\left(v^2 - \frac{\mu}{r} \right) \mathbf{r} - \left(\mathbf{r} \bullet \mathbf{v} \right) \mathbf{v} \right]$$

C3 is the "twice specific" orbital energy which is determined from the position **r** and velocity **v** vectors according to

$$C_3 = \left| \mathbf{v} \right|^2 - \frac{2\mu}{\left| \mathbf{r} \right|}$$

The right ascension and declination of the asymptote can be computed from components of the unit asymptote vector according to

$$\alpha_\infty = \tan^{-1}\left(s_x, s_y \right)$$

$$\delta_\infty = \sin^{-1}\left(s_z \right)$$

The syntax of the MATLAB function that performs these calculations is as follows.

```
function [c3, rla, dla] = rv2hyper (mu, rsc, vsc)

% convert position and velocity vectors to
% hyperbolic c3, rla and dla

% input

%  mu   = gravitational constant (km**3/sec**2)
%  rsc  = spacecraft position vector (kilometers)
%  vsc  = spacecraft velocity vector (kilometers/second)

% output
```

```
%  c3  = specific orbital energy (km/sec)**2
%  rla = right ascension of asymptote (radians)
%  dla = declination of asymptote (radians)
```

Using classical orbital elements

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

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

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

The declination of the asymptote (DLA) is given by

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

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

From the following two expressions

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

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

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

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

Please note that the inverse tangent in these expressions is a four quadrant calculation.

The syntax of the MATLAB function that performs these calculations is as follows.

```
function [shat, rasc_asy, decl_asy] = orb2hyper(oev)

% this function converts classical orbital elements
% of a hyperbolic orbit to asymptote coordinates

% input
```

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

% output

%  shat     = asymptote unit vector
%  rasc_asy = right ascension of asymptote (radians)
%  decl_asy = declination of asymptote (radians)
```

Algorithm resources

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

Spacecraft Mission Design, Charles D. Brown, AIAA Education Series, 1992.

Orbital Mechanics, Vladimir A. Chobotov, AIAA Education Series, 2002.