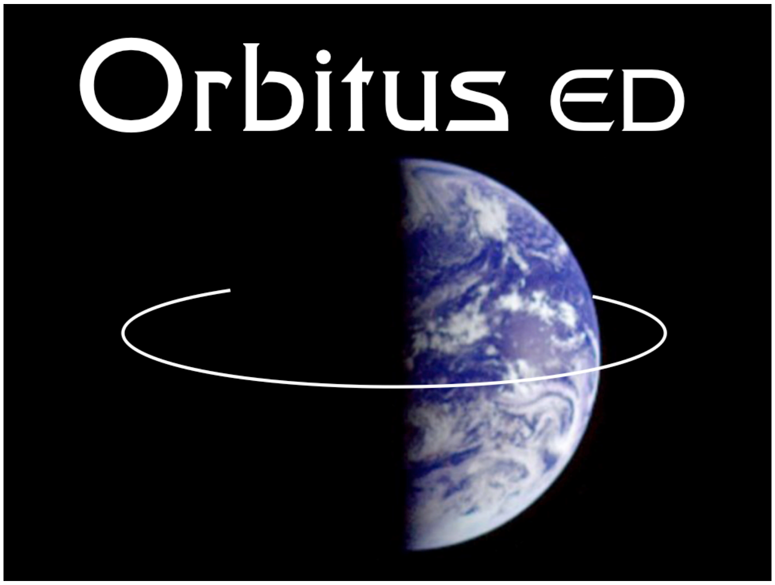


Orbit Mechanics and Manipulation Tool

User Manual

Version 1.2

Ben K. Bradley and Suneel I. Sheikh

3 May 2013

www.asterlabs.com

Contents

Overview

Orbitus ED is a graphical user interface (GUI) written in MATLAB accomplished at displaying satellite orbits, orbit transfers, rendezvous, as well as generating usable data associated with these trajectories and maneuvers.

This software is a perfect tool for engineering, physics, and astronomy students and professors of orbital mechanics. It is also an excellent tool for use in industry for quick visualizations and analysis of satellite orbits. The user interface is intuitive and powerful. It takes only seconds to generate a satellite ground track or trajectory with the stunning graphics usually reserved for expensive commercial tools.

System Requirements

Orbitus ED is designed to work within MATLAB version 7.5 (R2007b) and higher.

MATLAB versions earlier than 2007b are not supported due to changes in *.fig files and pre-parsed pseudocode, or p-code, encryption that MATLAB has changed over the years.

However, if you have a strong need of using versions earlier than MATLAB version 7.5 for your application, please contact our support group for further assistance:
support@asterlabs.com

Orbitus ED will work on any platform, including Mac OS, Windows, or Linux, which supports the MATLAB versions listed above.

MATLAB® is a high-level language and interactive development environment.
It is produced by The MathWorks, Inc.
Users are referred to their website for more information:
http://www.mathworks.com/products/matlab/

Installation

(1) Download the zipped folder from the ASTER Labs web store after your purchase:
http://www.asterlabs.com/store

(2) If not done automatically by your web browser, unzip the downloaded folder.

(3) Put the unzipped Orbitus ED folder into a directory or folder of your choice.

(4) Open up MATLAB and navigate to the Orbitus ED folder.

(5) At the MATLAB command line, type the following command then hit enter:

```
>> Install_OrbitusED
```

(6) Follow any instructions shown in the installation user interface that appears.

(7) Orbitus ED is now installed. A message should appear in the command window with a summary of the Orbitus ED license information.

(8) This Orbitus ED license is tied to the MATLAB license number that was entered when the software was purchased.

(9) To start Orbitus ED, type the following in the command window and hit enter:

```
>> OrbitusED
```

Important Notes

(a) The value for the Earth's gravitational parameter being used is 398,600.4415 km^3/s^2

(b) The value for the Earth's radius being used is 6,378.137 km

(c) External orbit perturbations are not taken into account in any part of the software

(d) All three-dimensional plots can be rotated and zoomed

(e) If user interface windows appear bigger than your screen, you may need to change the resolution of your monitor.

(f) Information about your Orbitus ED license and version can be viewed by entering the following at the MATLAB command line:

```
>> About_OrbitusED
```

(g) Known Issue: The MATLAB function of `gui_mainfcn` can occasionally produce an error to the command window when switching between Orbitus ED modules. This is known only to happen in MATLAB version 7.5 (R2007b) and earlier. However, this GUI error does not affect the Orbitus ED module function or computations.

(h) We would happily take suggestions or noted corrections. Please submit these directly to our support group via email at:
support@asterlabs.com

What's New in Version 1.2?

There are some great new features that have been added to Orbitus ED for this new release, including the ability for the user to make use of the built-in functions of Orbitus ED for their own endeavors. A summary of the updates is provided below:

- All the MATLAB functions of Orbitus ED are now available for the user to utilize on their own and interface them with their own code! A new subfolder called "UserFunctions" contains these usable functions. Each function has an accompanying m-file that contains everything the user needs to know in order to use the function correctly and effectively. This information can be accessed by either opening the m-file or typing

  ```
          >> help function_name
  ```

  at the command line of MATLAB, where function_name is the actual name of the script. A description of each available function is provided in the **User Functions** section of this manual.

- The MATLAB Image Processing Toolbox is no longer needed to display the high quality central body maps! The standard MATLAB toolboxes now provide this capability.

- The user accessible function `CentralBody_Plot` now allows the user to plot the Sun, Earth, Moon, Venus, Mars, or Jupiter as a central body with a high quality map of that body. See the **User Functions** section of this manual for more information.

- A number of functions have had their processing sped up through increased vectorization of the code.

Software Modules

MAIN MENU

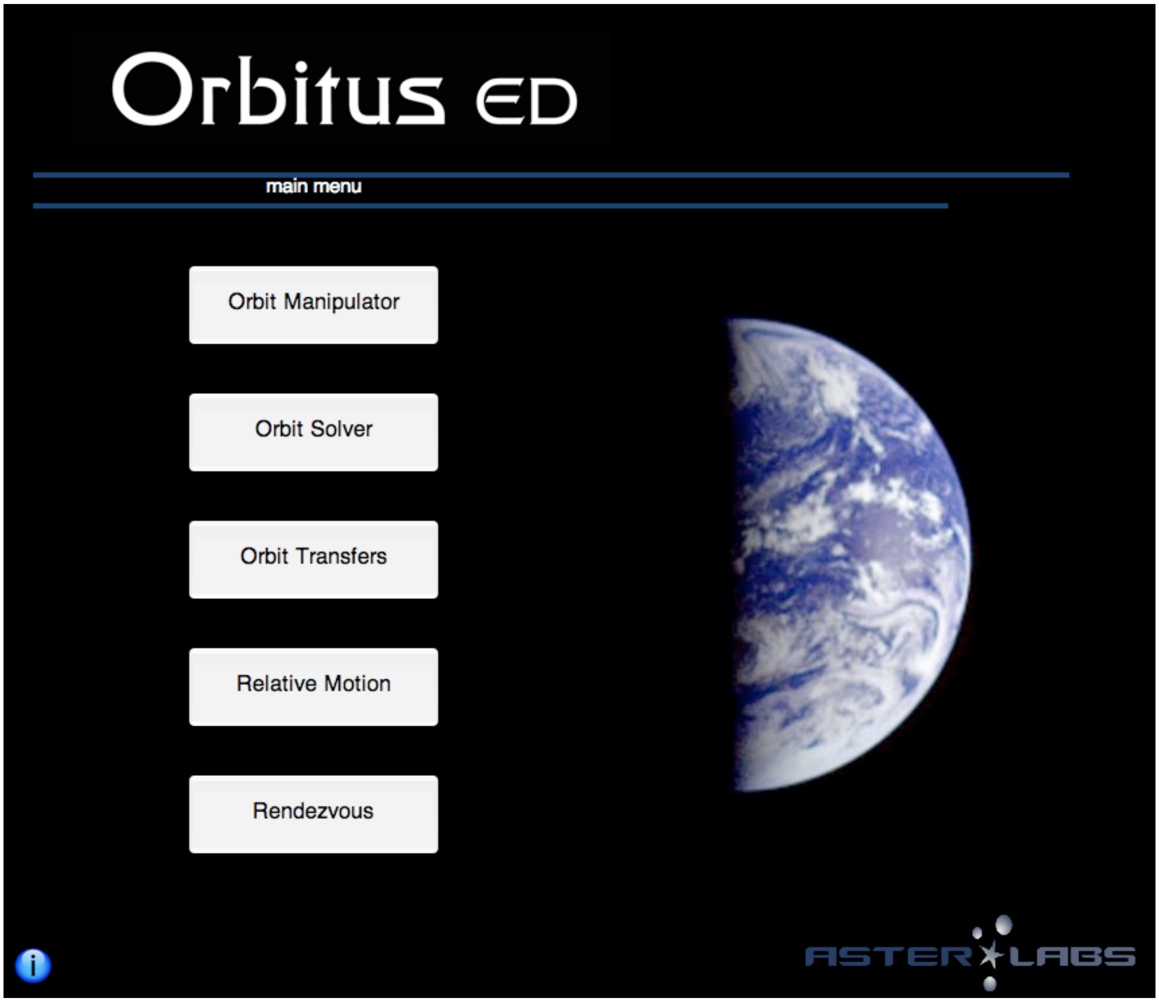


Figure 1. *Main Menu* module of Orbitus ED

Figure 1 shows the *Main Menu*, which is the first screen that the user views after starting Orbitus ED. The *Main Menu* allows the user to select the type of orbital mechanics problem that they would like to visualize and solve. Once a problem type is selected, the *Main Menu* disappears and a window corresponding to the user selection appears.

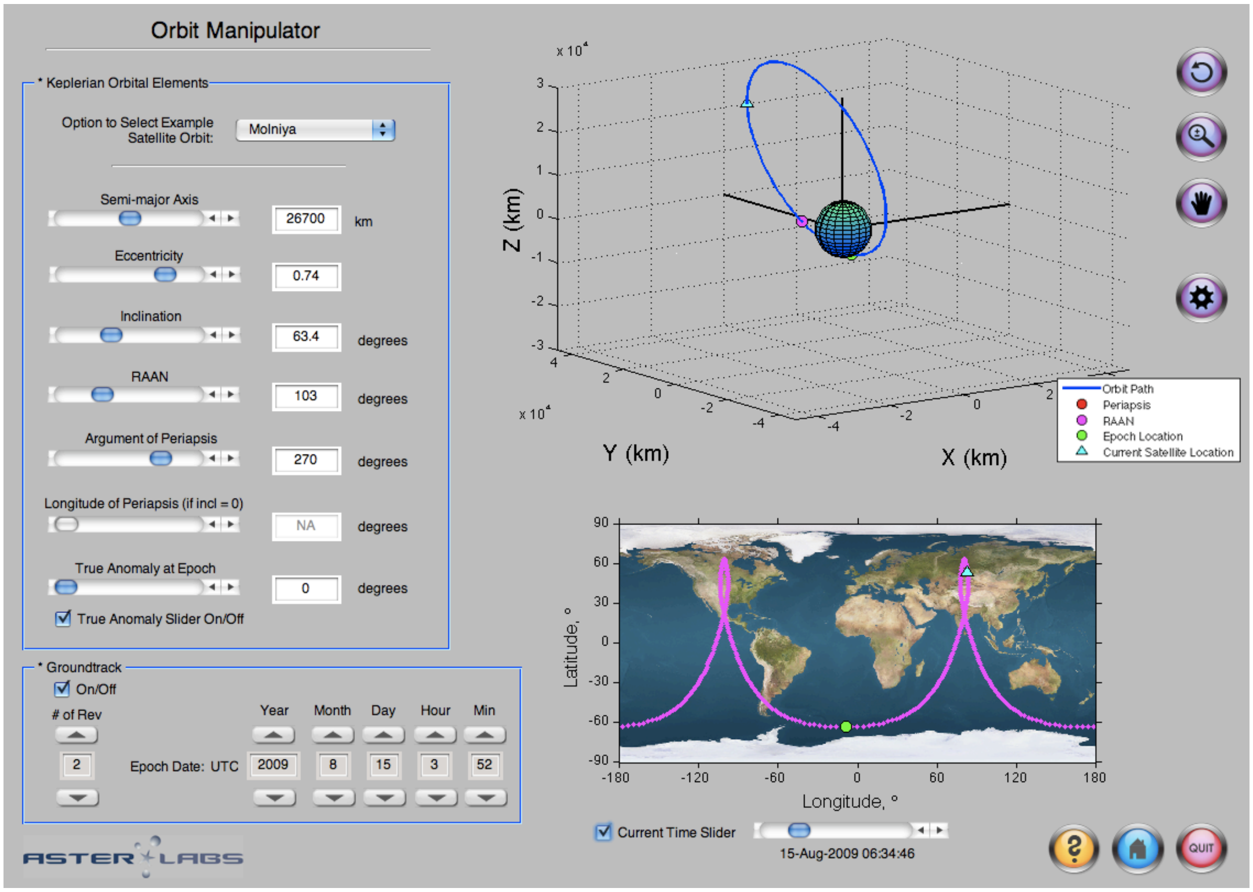
There are five modules to choose from:

- *Orbit Manipulator* : Allows the user to modify and visualize various orbits
- *Orbit Solver* : Converts between Keplerian and Position & Velocity type orbit elements
- *Orbit Transfers* : Computes Hohmann and Bi-elliptic orbit transfers
- *Relative Motion* : Computes information of the relative motion between a primary and secondary spacecraft in near proximity to the primary vehicle
- *Rendezvous* : Given rendezvous duration, computes energy and path required to rendezvous with another orbit

On the *Main Menu* an *Info* button is provided in the lower left corner. Clicking this button, labeled 'i', opens the *About* page. This page provides the *Software version number,* as well as *Special Thanks/References* and the *Terms of Use*. The *Terms of Use* in PDF format is also included in the downloaded zipped folder. Additionally, you can check for software updates from this page.

Each module of the software has three yellow, blue, and red buttons located in the lower right corner of each GUI window. The yellow button with a question mark is an *Info* button. Clicking this button brings up a new window with information on the type of problem that is to be solved or plotted, as well as definitions of some relevant key terms. The blue button with a home symbol in it will take the user back to the *Main Menu* of Orbitus ED. The red button with *QUIT* inside it will, that's right you guessed it, quit the program if clicked. Don't worry, this button won't quit MATLAB. It only closes the Orbitus ED GUI window.

ORBIT MANIPULATOR



Orbit Manipulator

* Keplerian Orbital Elements

Option to Select Example Satellite Orbit: Molniya

Semi-major Axis: 26700 km

Eccentricity: 0.74

Inclination: 63.4 degrees

RAAN: 103 degrees

Argument of Periapsis: 270 degrees

Longitude of Periapsis (if incl = 0): NA degrees

True Anomaly at Epoch: 0 degrees

☑ True Anomaly Slider On/Off

* Groundtrack

☑ On/Off

# of Rev: 2

Epoch Date: UTC

Year	Month	Day	Hour	Min
2009	8	15	3	52

☑ Current Time Slider

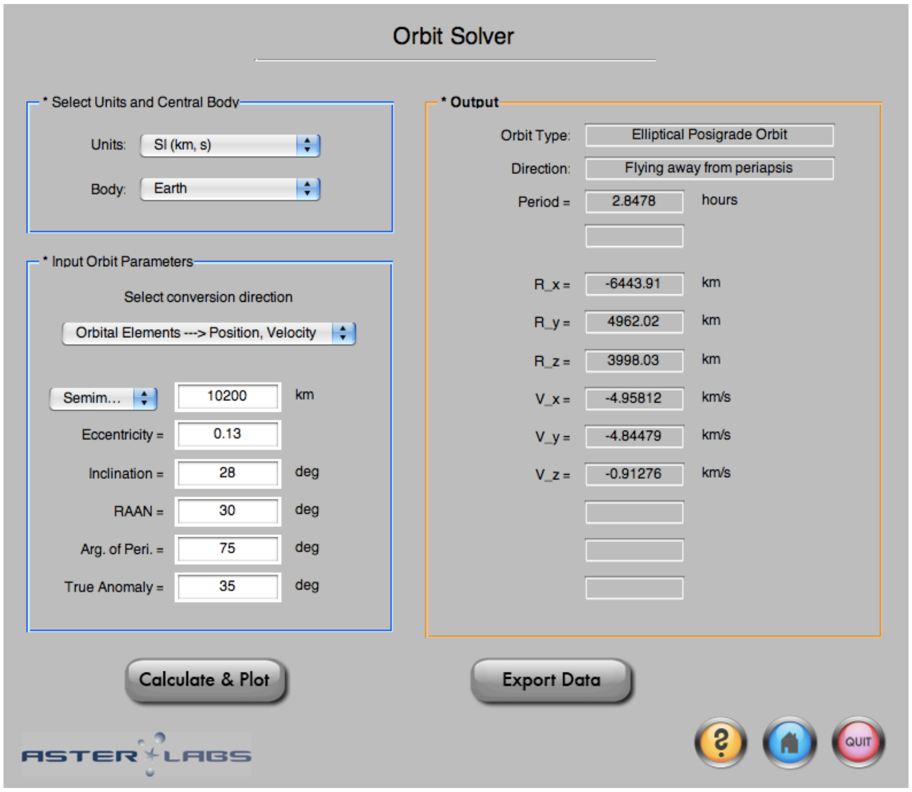
15-Aug-2009 06:34:46

Figure 2. *Orbit Manipulator* module.

The *Orbit Manipulator* module allows the user to manipulate the Keplerian orbital elements of an orbit using slide bars on the left-hand side of the window. As any of these orbit elements are modified, the picture of the orbit on the right-hand side changes in real-time. The image of the orbit in 3D is plotted in the Earth-Centered Inertial (ECI) frame. The Plot Settings button to the right of the 3D plot (shaped like a gear) allows the user to plot the apse line, node line, and the equatorial plane. Utilizing these options can aide in the understanding of the orbital elements' effect on the orbit.

Additionally, the Groundtrack input box in the lower left allows the user to define the date and time at which the orbital elements were observed. A groundtrack trace is plotted to the lower right in a two-dimensional map display. When the groundtrack is turned on, any changes to the orbital elements will also update the groundtrack display. The slidebar beneath the groundtrack controls the current time (in UTC), thus controlling the current position of the satellite in its orbit.

ORBIT SOLVER



Orbit Solver

* Select Units and Central Body

Units: SI (km, s)

Body: Earth

* Input Orbit Parameters

Select conversion direction

Orbital Elements ---> Position, Velocity

Semim... 10200 km

Eccentricity = 0.13

Inclination = 28 deg

RAAN = 30 deg

Arg. of Peri. = 75 deg

True Anomaly = 35 deg

* Output

Orbit Type: Elliptical Posigrade Orbit

Direction: Flying away from periapsis

Period = 2.8478 hours

R_x = -6443.91 km

R_y = 4962.02 km

R_z = 3998.03 km

V_x = -4.95812 km/s

V_y = -4.84479 km/s

V_z = -0.91276 km/s

Figure 3. *Orbit Solver* module.

The *Orbit Solver* module gives the user the ability to plot any orbit and convert between Keplerian orbital elements and a position and velocity vector in the ECI frame. Additionally, the Sun can be selected as the central body. In this case, position and velocity would be in the Heliocentric Coordinate frame.

Once an orbit is plotted, the user can export the position and velocity tabulated data within the orbit to either a text file or Microsoft Office Excel file. This gives the user the opportunity to examine certain properties of the orbit in further detail in any manner they desire.

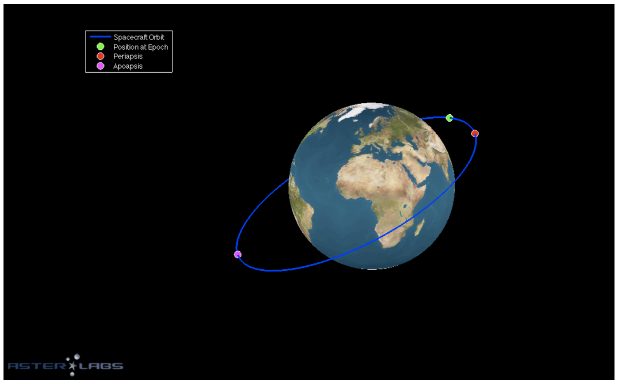


Figure 4. Elliptical orbit created by *Orbit Solver* module.

ORBIT TRANSFERS

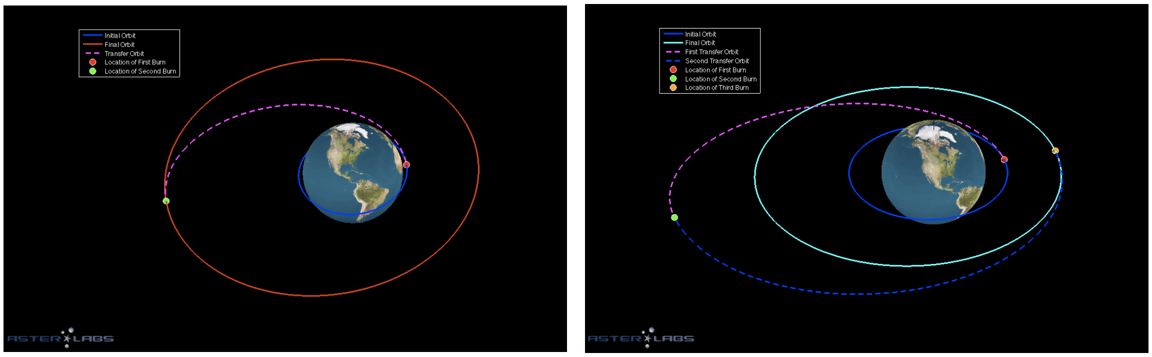
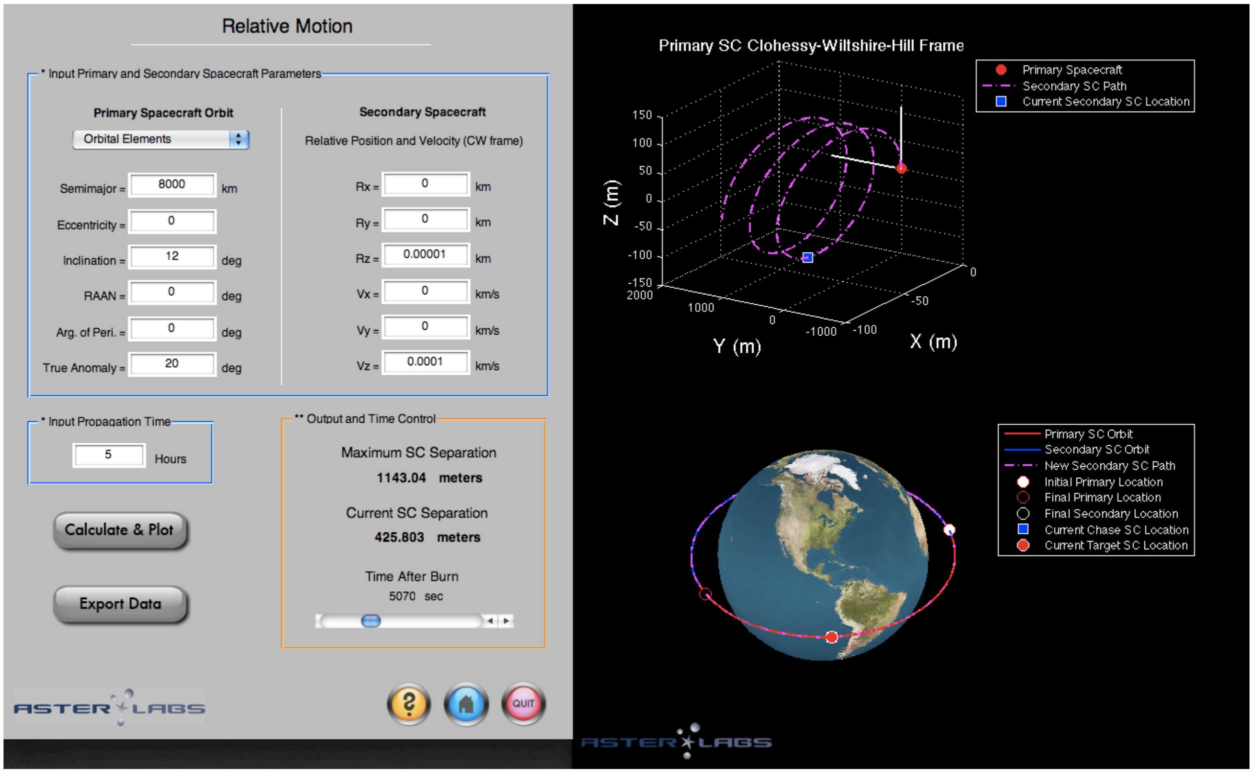


Figure 5. Left: Hohmann transfer. Right: Bi-elliptic transfer. Created by the two *Orbit Transfer* modules.

Orbital maneuvers can transfer a spacecraft from one orbit to another. Using the *Orbit Transfers* module of the Orbitus ED software allows the user to calculate and plot Hohmann transfers and bi-elliptic transfers. For Hohmann transfers, the user only needs to specify the initial and final orbit elements of the transfer. Bi-elliptic transfers require the user to also include either a maximum intermediate radius or duration of the transfer. An example of an intermediate radius is illustrated using a green trace in the right-hand plot of Figure 5.

The orbits may be specified by orbital elements, position and velocity vectors, or simply combinations of eccentricity and periapsis or apoapsis. Once plotted, the required incremental velocity, or *delta-v*, information is displayed in the GUI window and the position and velocity data, as well as the delta-v data can be exported. Additionally, the user has the option of using either Earth or the Sun as the orbit's central body.

RELATIVE MOTION



Relative Motion

* Input Primary and Secondary Spacecraft Parameters

Primary Spacecraft Orbit

Orbital Elements

Semimajor = 8000 km

Eccentricity = 0

Inclination = 12 deg

RAAN = 0 deg

Arg. of Peri. = 0 deg

True Anomaly = 20 deg

Secondary Spacecraft

Relative Position and Velocity (CW frame)

Rx = 0 km

Ry = 0 km

Rz = 0.00001 km

Vx = 0 km/s

Vy = 0 km/s

Vz = 0.0001 km/s

* Input Propagation Time

5 Hours

** Output and Time Control

Maximum SC Separation

1143.04 meters

Current SC Separation

425.803 meters

Time After Burn

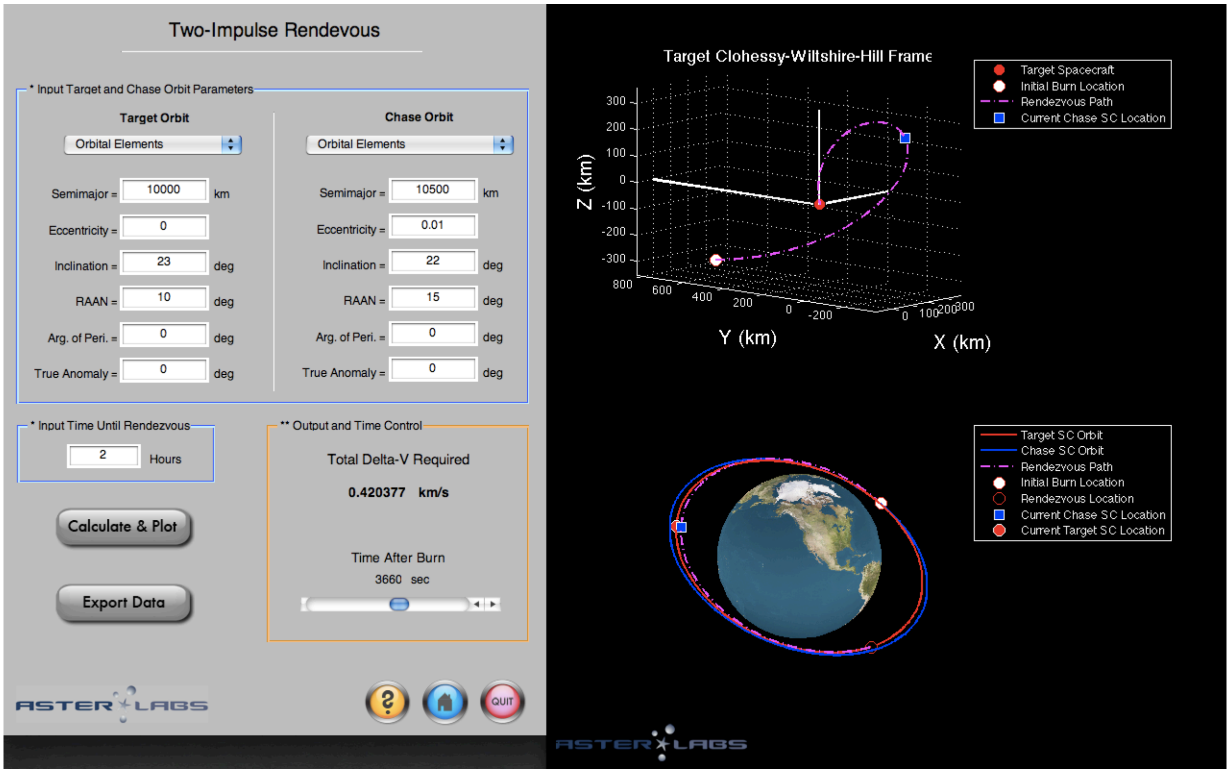
5070 sec

Figure 6. ***Relative Motion*** **module.**

To utilize the *Relative Motion* module, the user must specify three things: the primary spacecraft orbit, the secondary spacecraft relative position and velocity, and the amount of time desired to propagate the scenario into the future. The primary spacecraft orbit can be defined by orbital elements or a position and velocity vector in the ECI frame. The secondary spacecraft must be given a relative position and velocity vector in the Clohessy-Wiltshire-Hill frame. Once the *Calculate & Plot* button is pressed, two plots are generated to the right-hand side as shown in Figure 6. The plot on top displays the secondary spacecraft's path with respect to the primary spacecraft over the designated amount of time. This is in the Clohessy-Wiltshire-Hill frame. The plot on the bottom displays the two spacecraft's orbits in the ECI frame. It should be noted that the Clohessy-Wiltshire-Hill equations are being used for these calculations. This means that the eccentricity of the primary spacecraft orbit needs to be zero or close to zero in order to get valid solutions. Also, the initial relative position cannot be zero in magnitude. An easy fix is to make the initial position very small (a few mm or cm) if required.

Additionally, the maximum spacecraft separation distance and the current separation distance are displayed in the output box of the GUI window. A time slide bar controls the time since the initial conditions were valid. As the time is adjusted, the value in the *Current SC Separation* field is updated, as are the spacecraft positions in the plots. Orbit position and velocity data can also be exported for further investigation.

RENDEZVOUS



Two-Impulse Rendevous

* Input Target and Chase Orbit Parameters

	Target Orbit		Chase Orbit	
	Orbital Elements		Orbital Elements	
Semimajor =	10000	km	10500	km
Eccentricity =	0		0.01	
Inclination =	23	deg	22	deg
RAAN =	10	deg	15	deg
Arg. of Peri. =	0	deg	0	deg
True Anomaly =	0	deg	0	deg

* Input Time Until Rendezvous

2 Hours

** Output and Time Control

Total Delta-V Required

0.420377 km/s

Time After Burn

3660 sec

Figure 7. *Rendezvous* module.

The *Rendezvous* module facilitates the visualization of complex rendezvous of two spacecraft. To implement a rendezvous, the user must input the target spacecraft orbit, chase spacecraft orbit, and the desired duration of the rendezvous maneuver. The target spacecraft orbit can be specified by orbital elements or a position and velocity vector in the ECI frame. The chase spacecraft orbit can be specified by orbital elements, a position and velocity vector in the ECI frame, or relative position and velocity vectors in the Clohessy-Wiltshire-Hill frame. It should be noted that the Clohessy-Wiltshire-Hill equations are being used for these calculations. This means that the eccentricities of both orbits need to be zero or close to it in order to get valid solutions.

Once the *Calculate & Plot* button is pressed, two plots are generated to the right-hand side, as shown in Figure 7, and the required delta-v is displayed in the output box. The plot on top is the path of the chase satellite during the maneuver in the Clohessy-Wiltshire-Hill frame. The plot on the bottom consists of the target orbit, original chase orbit, and the path of the chase spacecraft during the rendezvous maneuver all plotted in the ECI frame. Additionally, there is a slide bar in the GUI window that controls the time after the initial burn. As time since the initial burn, the *Time After Burn*, slide bar is adjusted, the target and chase spacecraft update their locations on both plots. All orbit position and velocity data can also be exported for further research.

User Functions

Orbitus ED functions are available for the user to utilize as desired.

These functions are stored in the "UserFunctions" subfolder. Each encrypted function (each p-file) has an accompanying m-file containing a detailed header of important information on how to use the function. This information can be accessed by either opening the m-file or typing `>> help function_name` at the command line of MATLAB.

A list of available functions is given below. An example function header is provided in Figure 8 for the function `eccentric2true`.

```
%==========================================================================
%==========================================================================
% [true_anom] = eccentric2true(E, e)
%
% Calculates the true anomaly from eccentric anomaly. This works for all
%  orbit types (elliptic, parabolic, hyperbolic).
%
%
% Author: Ben K. Bradley
% Last Revision Date: 26 October 2010
%
% Copyright © 2008-2010 by ASTER Labs, Inc.
% All rights reserved.
%
%
% INPUT:           Description
Units
%
%  E            - Eccentric Anomaly, single value or vector             rad
%  e            - eccentricity of orbit
%
% OUTPUT:
%
%  true_anom    - True Anomaly, same size as input E            [0,2pi] rad
%
%
% Coupling:
%
%  none
%
% References:
%
%  [1] Vallado, D.A., "Fundamentals of Astrodynamics and Applications,"
%         Third Edition, Microcosm Press, 2007.
%
%  [2] Curtis, H.D., "Orbital Mechanics for Engineering Students,"
%          Elsevier Ltd., 2005.
%
%==========================================================================
%==========================================================================
```

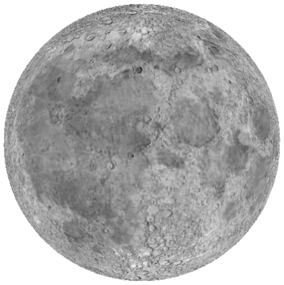
Figure 8. Example header for sample function (`eccentric2true`).

CentralBody_Plot

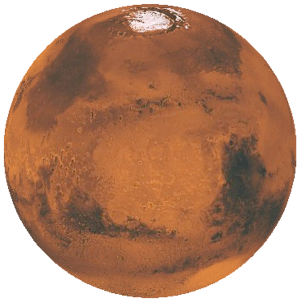
Plots the desired central body in a new figure window or an existing one if given the axes handle. Central body options are the Sun, Venus, Earth, Moon, Earth and Moon, Mars, and Jupiter. Additionally, units can be set to SI or canonical. Each body has a semi-major axis, semi-minor axis, and picture file associated with it. These items are given in the table below.

Body	Semi-major Radius [km]	Semi-minor Radius [km]	Image File
Sun	695,500	same	sun2.jpg
Venus	6,051.8	same	venus2.jpg
Earth	6378.137	6356.752	earth1.jpg
Moon	1738	same	moon2.jpg
Mars	3396.19	3376.2	mars3.jpg
Jupiter	71,492	66,854	jupiter1b.jpg

When the "Earth and Moon" option is input the Earth is placed at the center of the axes and the Moon's center of mass is placed at 384,400 km along the x-axis. The function outputs handles to each central body surface object, allowing the user to move and rotate each body.



Earth's Moon



Mars

Jupiter

datevec_dt

Calculates a new date/time vector from an original date/time vector and a time step in either days or seconds.

eccentric2true

Calculates the true anomaly from eccentric anomaly. This works for all orbit types (elliptic, parabolic, and hyperbolic). This function can handle a single eccentric anomaly or a vector of many values. The header for this function is shown in Figure 8.

eci2LatgcLon

Calculates geocentric latitude and longitude given a position in the Earth-Centered Inertial (ECI or GCRF) frame and UTC date/time. A simple approximation using Greenwich Mean Sidereal Time is used to transform the given ECI position to ECEF (Earth-Centered, Earth-Fixed).

For more precise computations, the geodetic latitude should be computed using the function `eci2LatgdLon`, which also calculates ellipsoidal height. Additionally, more accurate ECI/ECEF transformations can be computed using the IAU-2006/2000A method (see IERS Conventions 2010).

eci2LatgdLon

Calculates geodetic latitude and longitude given a position in the Earth-Centered Inertial (ECI or GCRF) frame and UTC date/time using an iterative method. A simple approximation using Greenwich Mean Sidereal Time is used to transform the given ECI position to ECEF (Earth-Centered, Earth-Fixed).

Tolerance = 1e-8
Constants (WGS84): Semi-major Axis of Earth = 6378.137 km
Semi-minor Axis of Earth = 6356.75231425 km

GMST_Calc

Calculates the Greenwich Mean Sidereal Time (degrees east of vernal equinox).

hohmann_transfer

Computes burn magnitudes and transfer ellipse parameters for a Hohmann transfer.

jd2utc_mjd

Converts a Julian date into a calendar date/time vector and modified Julian date.

Julian_Day

Calculates the Julian day given a date/time vector. Julian date is defined as the number of days since noon on January 1st, 4713 BC.

kepler

Calculates position and velocity vectors after an elapsed time, given the initial position and velocity vectors. This uses the universal formulation of Kepler's equation. The inputs may be specified in SI units or canonical units.

LeapYear_Check

Determines if the year given is a leap year (i.e. February has 29 days). Near-term leap years should be 1980, 1984, 1988, 1992, 1996, 2000, 2004, 2008, 2012, and 2016.

mean2eccentric

Calculates the eccentric anomaly from the mean anomaly using Newton's method. The tolerance is set to 1e-8.

OrbE2rv

Orbital elements are used to calculate position and velocity vectors in the Geocentric Equatorial Coordinate System (GECS or ECI). Works for elliptic, parabolic, and hyperbolic trajectories.

This has been vectorized to compute many positions simultaneously. The input for true anomaly may be a single value or a column vector of many values. Additionally, the desired value for the gravitational parameter may be input OR a character may be given to specify use of a stored value. For Earth, use 'E' (398600.4415 km^3/s^2). For Sun, use 'S' (1.32712428e11 km^3/s^2). If canonical units are desired use 'c' and the gravitational parameter will be set to 1.

rv2OrbE

Computes the Keplerian (classical) orbital elements (including special cases) given a position and velocity in the ECI (Earth-Centered Inertial) frame. Works for elliptic, parabolic, and hyperbolic trajectories.

The desired value for the gravitational parameter may be input OR a character may be given to specify use of a stored value. For Earth, use 'E' (398600.4415 km^3/s^2). For Sun, use 'S' (1.32712428e11 km^3/s^2). If canonical units are desired use 'c' and the gravitational parameter will be set to 1.

time_of_flight

Calculates the time of flight from periapsis to the desired eccentric anomaly. A single eccentric anomaly or vector of many values may be input.

true2eccentric

Calculates the eccentric anomaly from true anomaly. This function can handle a single value for true anomaly or a vector of many values. This function also works for all orbit types (elliptic, parabolic, and hyperbolic).

Examples

Several tutorial examples are provided below for a select set of Orbitus ED modules. Users are encouraged to play with the defined input parameters to visualize and understand how an orbit or maneuver is affected. Playing and learning is an important feature of Orbitus ED!

ORBIT SOLVER

(1) From the Orbitus ED main menu click on *Orbit Solver*

(2) Select *SI (km, s)* as the units and *Earth* as the central body from the drop down menus

(3) Select *Orbital Elements ---> Position, Velocity* from the menu for the conversion method

(4) Select *Semi-major Axis* as the first orbital element

(5) Enter in the following orbital elements:

Semi-major axis	**12000**
Eccentricity	**0.3**
Inclination	**35**
RAAN	**20**
Arg. of Periapsis	**20**
True Anomaly	**60**

(6) Click the *Calculate & Plot* button

(7) You should now see a new window with the orbit plotted around the Earth. There are rotate and zoom buttons at the top of this new figure window. Also, a description of the orbit and the position and velocity vectors in the ECI frame should be displayed in the output box in the GUI window

(8) If you choose, you could export the position data of one revolution of the orbit to a file by clicking on the *Export Data* button in the GUI window

(9) Now let's try plotting a hyperbolic trajectory quickly

(10) Change the *Semi-major Axis* value to **-40000**

(11) Change the *Eccentricity* to **1.4**

(12) Click on *Calculate & Plot*

(13) Great! You've just plotted a hyperbolic trajectory in no time at all

(14) Return to the main menu by clicking the blue button in the lower right of the GUI

BI-ELLIPTIC TRANSFER

(1) From the main menu click on *Orbit Transfers*

(2) In the new window click on *Bi-Elliptic*

(3) Select *SI (km, s)* as the units and *Earth* as the central body

(4) Select *Eccentricity and Periapsis* in the drop down menu of the *Input Initial Orbit* box

(5) Input the following values for the initial orbit:

Eccentricity	**0.1**
Periapsis	**9000**

(6) Select *Periapsis and Apoapsis* from the drop down menu of the *Input Final Orbit* box

(7) Input the following values for the final orbit:

Periapsis	**15000**
Apoapsis	**20000**

(8) Select *Max. Intermediate Radius* from the drop down menu of the *Input Restriction on Transfer* box

(9) Enter a value of **35000** for the *Max. Intermediate Radius*

(10) Click on the *Calculate & Plot* button

(11) You should see a new window with the initial orbit, final orbit, and transfer trajectories plotted around Earth. The transfer time, intermediate radius, and delta-v required for the transfer is also displayed in the GUI window.

RENDEVOUS

(1) From the main menu click on *Rendezvous*

(2) Select *Orbital Elements* for both the target and chase spacecraft orbit

(3) Enter in the following values for both sets of orbital elements:

	Target	Chase
Semi-major	**10000**	**10500**
Eccentricity	**0**	**0.01**
Inclination	**23**	**22**
RAAN	**10**	**15**
Arg. of Periapsis	**0**	**0**
True Anomaly	**0**	**0**

(4) In the *Input Time Until Rendezvous* box, enter **2** hours

(5) Click on *Calculate & Plot*

(6) A new window with the trajectories plotted should appear to the right.

(7) The slide bar in the *Output and Time Control* box is now active. This controls the time since the initial burn. Altering the time will adjust the locations of the spacecraft in the plots accordingly.

References

[1] Bates, R.R., Mueller, D.D., and White, J.E., *Fundamentals of Astrodynamics*, Dover Publications, 1971

[2] Battin, R.H., *An Introduction to the Mathematics and Methods of Astrodynamics*, American Institute of Aeronautics and Astronautics, 1999.

[3] Chobotov, V.A. Ed., Orbital Mechanics, American Institute of Aeronautics and Astronautics, 1991.

[4] Curtis, H., *Orbital Mechanics: For Engineering Students*, Butterworth-Heinemann, 2005.

[5] Danby, J.M.A., *Fundamentals of Celestial Mechanics*, Willmann-Bell, 1992.

[6] Escobal, P.R., *Methods of Orbit Determination*, Krieger Publishing, 1965.

[7] Montenbruck, O., and Gill, E., *Satellite Orbits: Models, Methods, Applications*, Springer-Verlag, Berlin, 2000.

[8] Moulton, F.R., *An Introduction To Celestial Mechanics*, Dover Publications, 1970.

[9] Prussing, J.E., and Conway, B.A., *Orbital Mechanics*, Oxford University Press, 1993.

[10] Vallado, D.A., *Fundamentals of Astrodynamics and Applications*, Microcosm Press, 2007

Enjoy!!

We hope you find our Orbitus ED software useful.

We would happily take suggestions or noted corrections. Please submit these directly to us via email at support@asterlabs.com.

Development Group

Orbitus ED is developed by ASTER Labs, Inc.

Our development team includes Ben K. Bradley and Suneel I. Sheikh.