NASA/TM—2018-219992

Strategic Center for Networking, Integration, and Communications Orbit Propagation Front-End Software Development

Nick Kuhl and Kai Perlmutter
The Ohio State University, Columbus, Ohio

Bryan Welch
Glenn Research Center, Cleveland, Ohio

November 2018

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Technical Report Server—Registered (NTRS Reg) and NASA Technical Report Server—Public (NTRS) thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers, but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., "quick-release" reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Fax your question to the NASA STI Information Desk at 757-864-6500

- Telephone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Program
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM—2018-219992

Strategic Center for Networking, Integration, and Communications Orbit Propagation Front-End Software Development

Nick Kuhl and Kai Perlmutter
The Ohio State University, Columbus, Ohio

Bryan Welch
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

November 2018

Acknowledgments

Nick Kuhl and Kai Perlmutter would like to thank their mentor, Bryan Welch; the other NASA Strategic Center for Engineering, Networking, Integration, and Communications (SCENIC) interns: Robert Bacon, Jack Green, Lindsey Jasper, and Noah Kontur; the SCENIC project manager, Carol Quinn; and the Space Communications and Navigation Summer Internship Project coordinators Tim Gallagher and Lindsay Hill.

Level of Review: This material has been technically reviewed by technical management.

Available from

NASA STI Program
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
703-605-6000

This report is available in electronic form at http://www.sti.nasa.gov/ and http://ntrs.nasa.gov/

Strategic Center for Networking, Integration, and Communications Orbit Propagation Front-End Software Development

Nick Kuhl and Kai Perlmutter
The Ohio State University
Columbus, Ohio 43210

Bryan Welch
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Summary

Orbit propagation is fundamental for space-based mission analysis, requiring software tools to predict the time-based positions and velocities of orbiting satellites. Many system analysts currently rely on commercial software, which is effective but expensive. NASA's Space Communications and Navigation (SCaN) Strategic Center for Networking, Integration, and Communications (SCENIC) project intends to provide these analysis capabilities by using a combination of internal and open-source software, allowing for a greater flexibility while maintaining low costs. Several routines were created to take different forms of user input to translate input into the standard set of orbital elements to be utilized within the current SCENIC analysis capabilities. These inputs range from two-line elements (TLEs) and Cartesian vectors, to coordinate transformations and Walker Delta constellations. These routines will be integrated into the SCENIC user interface (UI) in order to provide greater customization and flexibility for orbit propagation and other dynamic-analysis capabilities.

1.0 Introduction

The prediction of orbits is crucial to performing space mission analysis. Satellite orbit determination and propagation methods involve utilizing the position and velocity information of a satellite orbit in order to determine orbital elements. However, analytical solutions to orbit determination problems are only possible for the simplest problems. As a result, a combination of analytical and numerical solutions are necessary for problems that are more complex. Furthermore, satellite data often does not directly observe inertial position and velocity. Instead, data are often received from radar, telemetry, optics, or the Global Positioning System (GPS) (Ref. 1), which need to be converted into the orbital elements. The complexity of analytical orbit determination as well as varying inputs provided from different data sources necessitates the use of software for space system analysts to effectively predict and utilize spacecraft orbits.

Space system analysts performing orbital analysis often rely on commercial software that can reliably predict spacecraft orbits. Although commercial software can be both reliable and effective for mission purposes, these tools are often expensive and difficult to work with due to the changing needs of the user. They may only perform calculations for specific inputs, forcing the user to comply with the needs of the software. To combat these drawbacks, the Space Communications and Navigation (SCaN) Strategic Center for Networking, Integration, and Communications (SCENIC) plans to utilize a combination of

MATLAB® (MathWorks®) and JavaScript routines with NASA-developed open-source toolboxes to effectively replicate the functionalities needed of orbit propagation without relying on proprietary commercial software. SCENIC will decrease project costs, while enabling the flexibility of possible analyses with a high degree of customization for future mission analysis capabilities.

SCENIC has provided a verified and validated orbit propagation tool that can reliably propagate various satellite orbits; however, there is a growing need for added functionalities to provide the user with more options for inputting data via a web-based, user-friendly interface. Further development into user-friendly tools that add greater degrees of customization was the primary focus of this project. In the following section, the methodology of this endeavor is described. Section 3.0 details the results and new functionalities that will be added to SCENIC. Section 4.0 provides a conclusion and highlights the potential for future work.

2.0 Methodology

Creation of these specific routines was heavily influenced by the need to create very user-friendly methods as well as to provide readable and maintainable code. The process of creating a routine began by implementing the algorithm in MATLAB® along with the use of the Jet Propulsion Laboratory (JPL) Spacecraft Planet Instrument C-matrix Events (SPICE) Toolkit (Ref. 2), an open-source toolbox that contains many planetary constants, such as gravitational parameters and radii. The MATLAB® scripts were then implemented in a front-end routine by using JavaScript and the Angular web-development framework.

Each specific routine developed needed to be accurately documented and tested. Workflows were created to give the reader a better understanding of the logic behind the routine. Unit tests were also developed for each individual routine to provide extensive test cases. In addition, each function contained detailed headers and extensive commenting throughout the code to reduce sources of confusion for future developers reading the code. Function headers include a function description, input and output arguments with units, other files needed to run the function, and author and revision information. Meaningful function and variable names were used to further reduce sources of confusion for future readers.

Next, the JPL Horizons On-Line Ephemeris System (Ref. 3) provided the capability to validate the routines by comparing the results to real-world data. After extensive documentation, verification, and validation, these routines were provided to the SCENIC development team for their integration into the SCENIC web-based user interface (UI) to improve the user's experience with the orbit propagation tool. Following this process is in line with NASA–STD–7009A, Standard for Models and Simulations (Ref. 4).

3.0 Results

The routines created as part of this project are described in this section. Several of the routines implemented as part of this project are used to convert specific input into the orbital elements used by the SCENIC UI, which are the semimajor axis (SMA), eccentricity, inclination, argument of periapsis, mean anomaly, and right ascension of the ascending node (RAAN).

3.1 Two-Line Element (TLE) Sets

A routine to convert TLE sets into the six orbital elements used by SCENIC was implemented. A TLE set is a data format that is used to describe the orbit of a geocentric satellite. Figure 1 shows an example of a TLE set for a National Oceanic and Atmospheric Administration (NOAA) weather satellite.

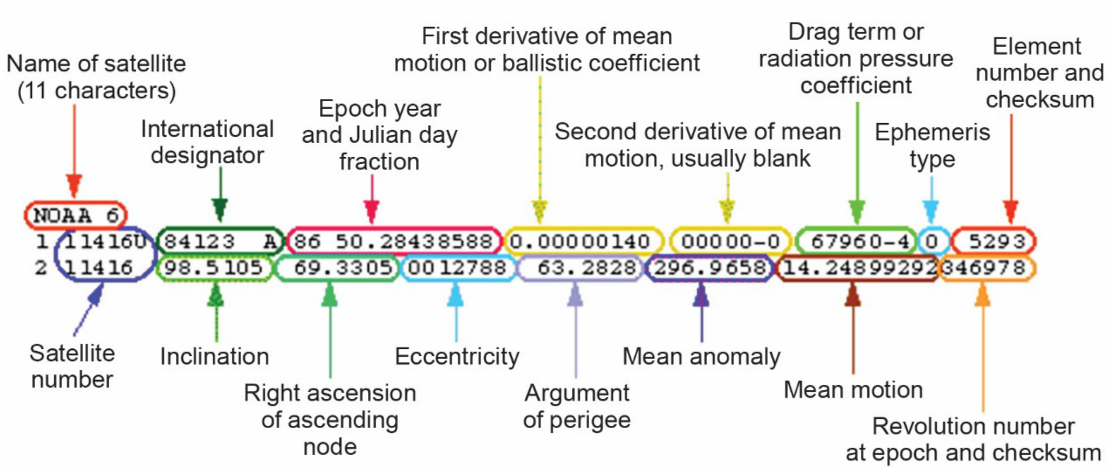


Figure 1.—National Oceanic and Atmospheric Administration (NOAA) weather satellite two-line element example (Ref. 5).

TLEs explicitly contain all six orbital elements used by SCENIC with the exception of the SMA. However, SMA a can be calculated in units of kilometers from the Earth's gravitational constant, μ, and the mean motion, n, of the satellite by using Equation (1).

$$a = \frac{\sqrt[3]{\dfrac{\mu}{\left(\left(n*2*\pi\right)/\left(24*3{,}600\right)\right)^2}}}{1{,}000} \tag{1}$$

Both the MATLAB® and front-end JavaScript versions allow the user to input as many TLE sets as they need. The user can then choose from a list to view the desired satellite's orbital elements. Figure 2 shows an example of the output for the MATLAB® version using a TLE set from the CelesTrak (Ref. 6) online database. Figure 3 shows a screenshot of the corresponding UI built with JavaScript and Angular.

The TLE routine has many features to make it as user-friendly as possible. The routine provides error messages to the user if the satellite numbers do not match between lines or if a checksum is not correct for a line. In addition, the routine allows the user to input TLE sets that have many formatting mistakes, such as blank lines, leading and trailing spaces, or lines that are backward.

3.2 Cartesian State Vectors

JPL Horizons provides the Cartesian state vectors for a desired satellite orbit at a specific date and time. JPL Horizons provides x, y, and z position and velocity for any target body, which can be extracted and used to determine the six orbital elements for SCENIC. Figure 4 shows JPL state vector data for the International Space Station. The output shows the state vectors as well as the date and time of the observation.

Field	Value
name	'ISS (ZARYA)'
LaunchYear	1998
LaunchNumber	67
Epoch	"10-Jun-2018 20:25:03"
SMA	6.7827e+03
e	3.3380e-04
i	51.6418
RAAN	50.3007
AP	171.6979
MA	280.7366

Figure 2.—MATLAB® two-line element output example, where AP is argument of periapsis, MA is mean anomaly, RAAN is right ascension of ascending node, and SMA is semimajor axis.

ISS (ZARYA) ▾

Parameter	**Value**
SMA (km)	6782.7232375089825
Eccentricity	0.0003338
Inclination (deg)	51.6418
Argument of Perigee (deg)	171.6979
Mean Anomaly (deg)	280.7366
RAAN (deg)	50.3007
Epoch (GMT)	10-Jun-2018 20:25:03

Enter a TLE File in the box below

```
ISS (ZARYA)
1 25544U 98067A   18161.85073725  .00003008  00000-0  52601-4 0  9993
2 25544  51.6418  50.3007 0003338 171.6979 280.7366 15.54163173117534
FLOCK 2E-1
1 41483U 98067JD  18161.93576331  .00706045  28730-3  64060-3 0  9998
```

Enter

Figure 3.—Two-line element (TLE) front-end interface, where RAAN is right ascension of ascending node, and SMA is semimajor axis.

```
2458266.500000000 = A.D. 2018-May-28 00:00:00.0000 TDB
 X = 8.297899703279406E-06 Y = 3.331279402736319E-05 Z =-2.968862351039235E-05
 VX=-3.113550724485021E-03 VY= 2.485383455223188E-03 VZ= 1.919678412331006E-03
 LT= 2.621354853851824E-07 RG= 4.538735232790110E-05 RR=-7.375216440131198E-07
2458267.500000000 = A.D. 2018-May-29 00:00:00.0000 TDB
 X =-1.117211580930243E-06 Y =-3.968535244899520E-05 Z = 2.181351000397611E-05
 VX= 3.087899693659367E-03 VY=-1.601815632234961E-03 VZ=-2.746420826171595E-03
 LT= 2.616255350133097E-07 RG= 4.529905715808112E-05 RR= 4.629289689035542E-06
2458268.500000000 = A.D. 2018-May-30 00:00:00.0000 TDB
 X =-5.107954983848083E-06 Y = 4.347769829837882E-05 Z =-1.192613256360919E-05
 VX=-2.924274687128069E-03 VY= 5.514251792926327E-04 VZ= 3.276315984919250E-03
 LT= 2.620478174475600E-07 RG= 4.537217309504417E-05 RR=-3.570860809515425E-06
```

Figure 4.—Jet Propulsion Laboratory Cartesian vector data for International Space Station from May 28 to 30, 2018 (Ref. 3).

A MATLAB® routine was created to convert the Cartesian vectors into orbital elements. In order to convert the data, the angular momentum vector, **h**, was first calculated from the position and velocity vectors **r** and **v**, respectively. The angular momentum vector was then used to determine the eccentricity vector, **e**, and the orbital inclination, *i*, which were determined by using Equations (2), (3), and (4)

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{2}$$

$$\mathbf{e} = \frac{\mathbf{v} \times \mathbf{h}}{\mu} - \frac{\mathbf{r}}{\|\mathbf{r}\|} \tag{3}$$

$$i = \cos^{-1} \frac{h_z}{\|\mathbf{h}\|} \tag{4}$$

where h_z is the angular momentum in the *z*-direction.

Next, the vector pointing toward the ascending node, $\boldsymbol{\eta}$, was determined and then used to calculate the RAAN, Ω, and the argument of periapsis, *w*. Equations (5), (6), and (7) were implemented in the routine to complete these calculations, where e_z is the eccentricity in the z direction, η_x is pointing toward the ascending node in the *x*-direction, and η_y is pointing toward the ascending node in the *y*-direction.

$$\boldsymbol{\eta} = (0,0,1)^T \times \mathbf{v} \tag{5}$$

$$\Omega = \begin{cases} \cos^{-1} \frac{\eta_x}{\|\boldsymbol{\eta}\|} & \text{for } \eta_y \geq 0 \\ 2\pi - \cos^{-1} \frac{\eta_x}{\|\boldsymbol{\eta}\|} & \text{otherwise} \end{cases} \tag{6}$$

$$w = \begin{cases} \cos^{-1}\dfrac{\langle \eta, \mathbf{e} \rangle}{\|\boldsymbol{\eta}\|\|\mathbf{e}\|} & \text{for } e_z \geq 0 \\ 2\pi - \cos^{-1}\dfrac{\langle \eta, \mathbf{e} \rangle}{\|\boldsymbol{\eta}\|\|\mathbf{e}\|} & \text{otherwise} \end{cases} \tag{7}$$

The true anomaly, θ, was then calculated in order to determine the eccentric anomaly, *E*, and finally the mean anomaly, *M*, of the orbit. Equations (8), (9), and (10) were utilized to implement these calculations.

$$\theta = \begin{cases} \cos^{-1}\dfrac{\langle \mathbf{e}, \mathbf{r} \rangle}{\|\mathbf{e}\|\|\mathbf{r}\|} & \text{for } \langle \mathbf{r}, \mathbf{v} \rangle \geq 0 \\ 2\pi - \cos^{-1}\dfrac{\langle \mathbf{e}, \mathbf{r} \rangle}{\|\mathbf{e}\|\|\mathbf{r}\|} & \text{otherwise} \end{cases} \tag{8}$$

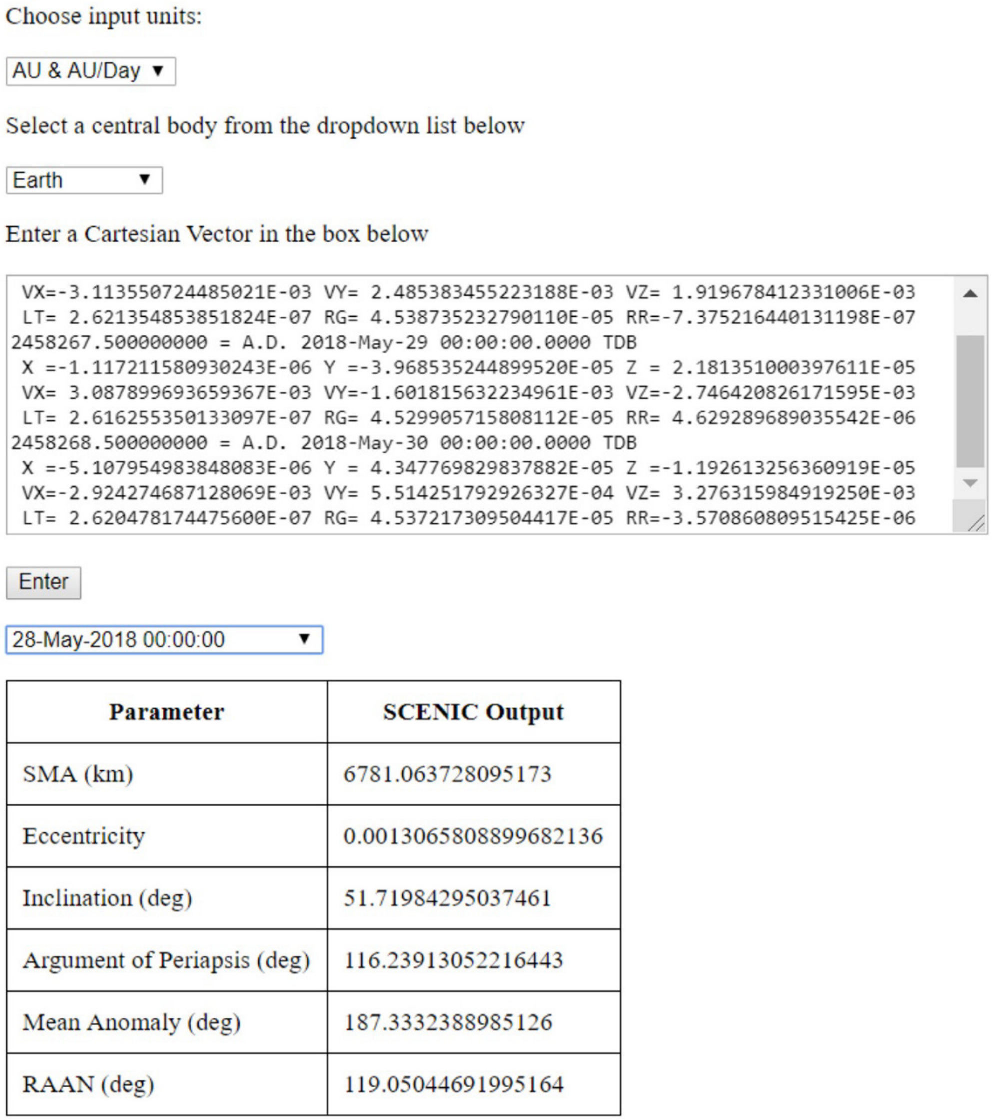
$$E = 2\tan^{-1}\frac{\tan\theta}{\sqrt{\dfrac{1+\|\mathbf{e}\|}{1-\|\mathbf{e}\|}}} \tag{9}$$

$$M = E - \|\mathbf{e}\|\sin E \tag{10}$$

Lastly, the SMA, *a*, was determined by using Equation (11). With the six orbital elements (SMA, eccentricity, inclination, RAAN, argument of periapsis, and mean anomaly), the results were outputted for the user.

$$a = \frac{1}{\dfrac{2}{\|\mathbf{r}\|} - \dfrac{\|\mathbf{v}\|^2}{\mu}} \tag{11}$$

After the MATLAB® routine to determine orbital elements from the Cartesian state vectors was created, a unit verification test with multiple test cases was created in order to compare outputs to the corresponding orbital elements from the JPL Horizons database. Additionally, a front-end routine built with JavaScript and Angular was created that performs the same task, but also accounts for the epoch date and time as well as the input units for each data point from the JPL Horizons database. A sample output of the front-end interface is shown in Figure 5, taking the input from Figure 4 and outputting the orbital elements.



Parameter	SCENIC Output
SMA (km)	6781.063728095173
Eccentricity	0.0013065808899682136
Inclination (deg)	51.71984295037461
Argument of Periapsis (deg)	116.23913052216443
Mean Anomaly (deg)	187.3332388985126
RAAN (deg)	119.05044691995164

Figure 5.—Angular interface for Cartesian vectors to orbital elements, where RAAN is right ascension of ascending node, SCENIC is Space Communications and Navigation Center for Engineering, Networking, Integration, and Communications, and SMA is semimajor axis.

3.3 Equatorial to Ecliptic Coordinate System

Spaceflight analysis heavily depends upon the coordinate system used to understand the position and motion of objects relative to the spacecraft. Selecting the right coordinate system can improve analysis capabilities as well as reduce the prospect for potential errors. A routine was created in order to obtain the ecliptic latitude and longitude of an object at a certain right ascension and declination. To do so, right ascension and declination data were obtained from JPL Horizons in the equatorial coordinate system, which were then converted into the ecliptic coordinate system in order to output the ecliptic longitude and latitude.

The equatorial coordinate system has an origin at the center of Earth with a primary direction toward the vernal equinox and a right-handed convention. Coordinates in the equatorial coordinate system are defined with a right ascension and declination. Figure 6 provides a visual for the equatorial coordinate system.

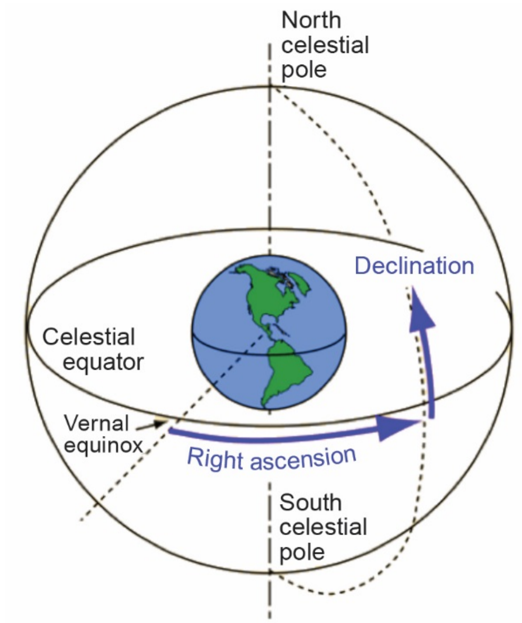


Figure 6.—Equatorial coordinate system (Ref. 7).

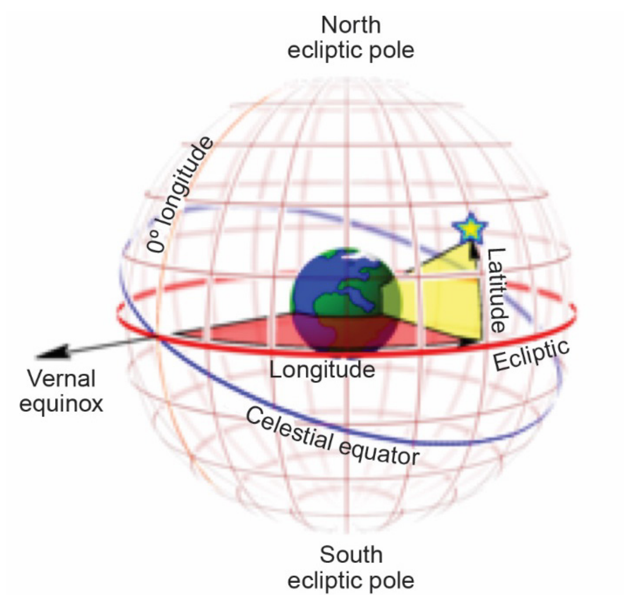


Figure 7.—Ecliptic coordinate system (Ref. 8).

The ecliptic coordinate system is centered either at the Sun or Earth and has a primary direction toward the vernal equinox. Unlike the equatorial coordinate system, the ecliptic system depends on longitude and latitude. A diagram of the ecliptic coordinate system is shown in Figure 7.

The routine to convert between equatorial and ecliptic coordinate systems started with taking right ascension α and declination δ input from JPL Horizons, and then calculating the ecliptic longitude, λ, and latitude, β, by using Equations (12) and (13). The axial tilt, ϕ, of the central bodies was hard coded in by using planetary fact sheets provided by Goddard Space Flight Center (Ref. 9).

$$\lambda = \tan^{-1}\frac{\sin\alpha\cos\delta + \tan\delta\sin\phi}{\cos\alpha} \tag{12}$$

$$\beta = \sin^{-1}(\cos\phi\sin\delta - \sin\alpha\cos\delta\sin\phi) \tag{13}$$

A MATLAB® routine for the coordinate transformation as well as a unit verification test were created and then implemented into an Angular framework by using JavaScript. The user inputs right ascension and declination as well as a central body in order to obtain latitude and longitude in the output. Figure 8 shows the outputs on the Angular interface for a sample test case.

3.4 Periapsis and Apoapsis Distances to Semimajor Axis (SMA) and Eccentricity

A routine to convert periapsis r_p and apoapsis r_a distances to SMA and eccentricity was also created. In an elliptical orbit, the periapsis is the closest point in the orbit and the apoapsis is the farthest point in the orbit. By using the periapsis and apoapsis altitude distances for a particular orbit, coupled with the radius of the central body, the SMA and eccentricity can be calculated by using Equations (14) and (15).

$$a = \frac{r_p + r_a}{2} \tag{14}$$

$$\mathbf{e} = \frac{r_a - r_p}{r_a + r_p} \tag{15}$$

A second MATLAB® routine was created, converting radius distances to SMA and eccentricity. The two routines are nearly identical, and both were combined into one front-end interface by using JavaScript. The user decides whether to input radius or altitude data and then selects a central body in order to receive an output on SMA and eccentricity. A sample output is shown in Figure 9.

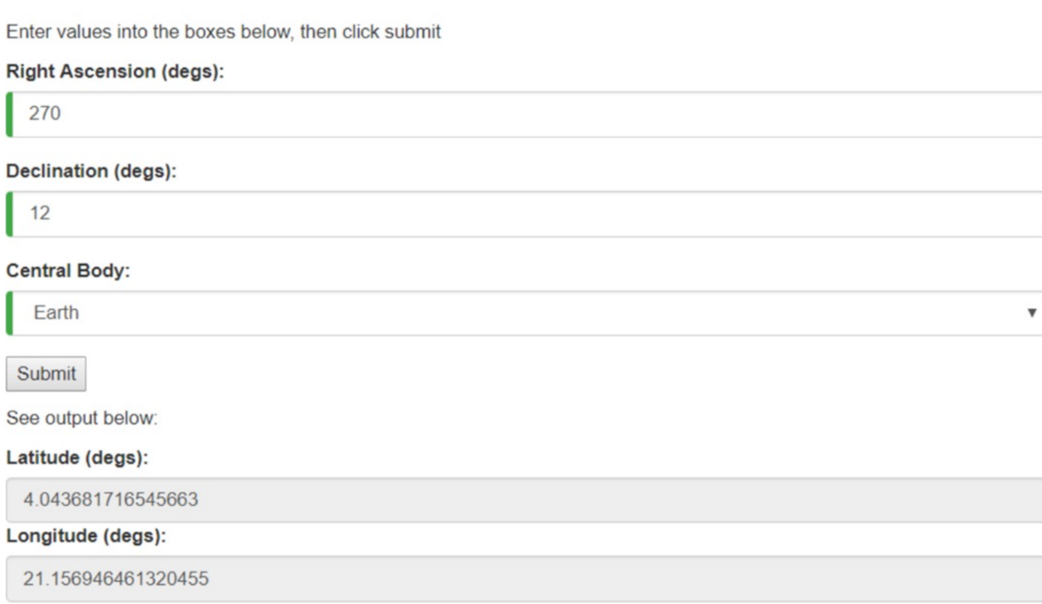


Figure 8.—Front-end output for equatorial to ecliptic transformation.

Figure 9.—Front-end output for altitude to semimajor axis and eccentricity routine.

3.5 Walker Delta Pattern Constellations

Satellite constellations are groups of satellites that work together to provide optimal coverage, dependent upon specific mission requirements. Satellites in the same constellation typically have similar eccentricity, inclination, and SMA. Walker Delta Pattern constellations are a common type of constellation with the notation i: $t/p/f$, where i is the inclination, t is the total number of satellites in the constellation, p is the number of orbital planes, and f is the phasing parameter (Ref. 1). A routine to output the orbital elements for each satellite in a Walker Delta constellation was created in both MATLAB® and JavaScript with the use of the Angular framework. The user inputs inclination, number of satellites, number of planes, the phasing parameter, and the SMA as well as values for the starting RAAN and starting mean anomaly. All satellites in a given Walker Delta constellation will have the same inclination and SMA, an eccentricity of 0 and an argument of periapsis of 0. Each satellite in a given plane will have the same RAAN, and the spacing between ascending nodes, Ω_Δ, of adjacent planes is calculated by using Equation (16).

$$\Omega_\Delta = 360/p \tag{16}$$

Satellites are equally spaced in each plane, and the mean anomaly spacing, M_Δ, between adjacent satellites is calculated by using Equation (17).

$$M_\Delta = 360/t/p \tag{17}$$

To calculate the phase difference between adjacent planes, Equation (18) is used.

$$f_\Delta = f * 360/t \tag{18}$$

A sample of the front-end interface built with JavaScript and Angular is shown in Figure 10. The user enters the relevant input information and then chooses the desired satellite’s orbital elements to display by using the drop-down list.

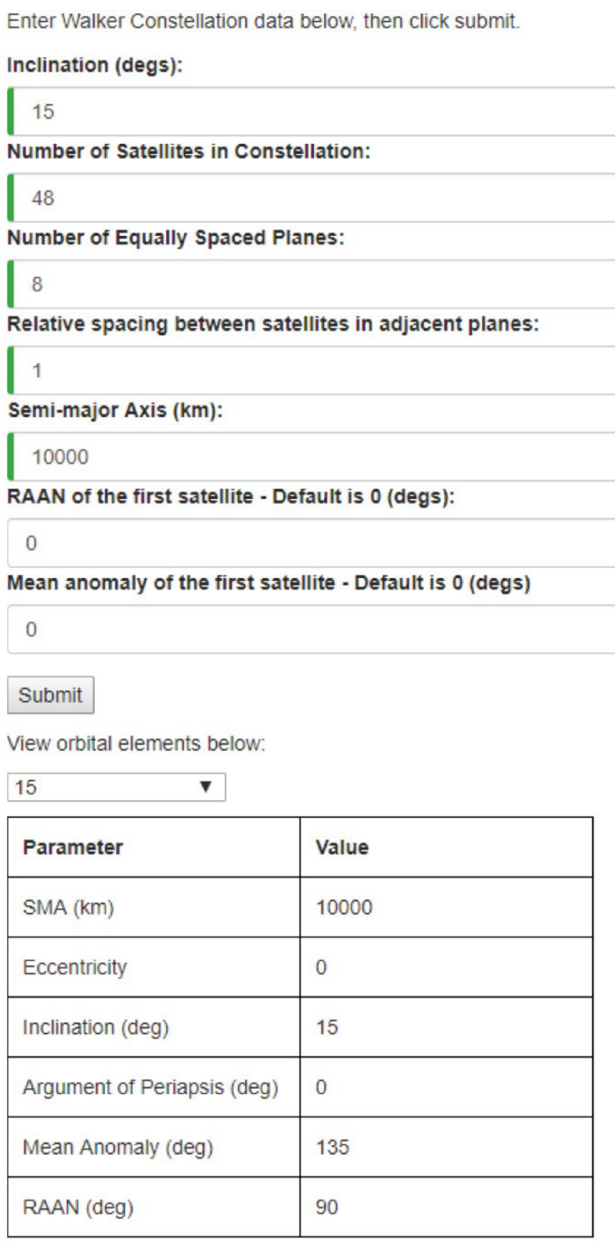


Figure 10.—Walker Delta constellation front-end interface, where RAAN is right ascension of ascending node, and SMA is semimajor axis.

In order to verify that the satellites in the output are evenly spaced, RAAN versus mean anomaly were plotted by using MATLAB®. Figure 11 shows a plot of RAAN versus mean anomaly for a 0: 48/8/1 Walker Delta constellation. Figure 11 shows that the satellites are evenly spaced out in the given constellation.

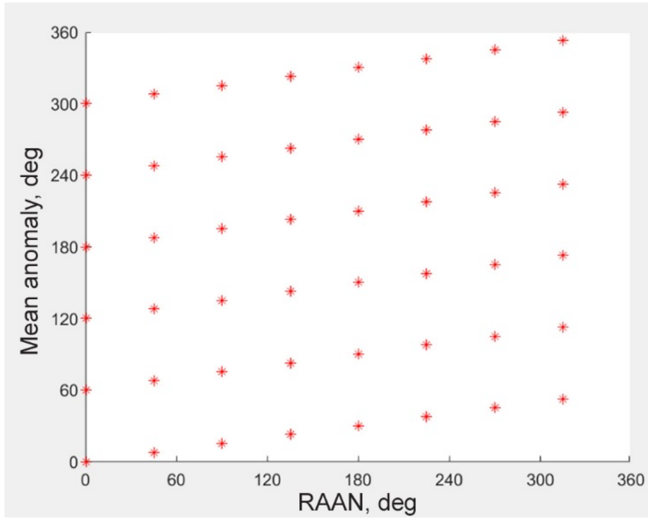


Figure 11.—Walker Delta constellation plot of right ascension of ascending node (RAAN) versus mean anomaly (0: 48/8/1).

4.0 Conclusion

Orbit propagation is an essential tool for performing time-dynamic capabilities such as line of sight, dynamic link analysis, dilution of precision navigation analysis, and orbital availability and latency calculations. The routines created as part of this project contribute toward the Space Communications and Navigation (SCaN) Strategic Center for Networking, Integration, and Communications (SCENIC) goals of providing a web-friendly user interface (UI), rapid flexibility, and a high level of customization. The front-end JavaScript routines support web-based flexibility and have increased the orbit propagation tool's level of customization by giving the user more options for entering orbital data.

There is work that still needs to be done regarding the SCENIC orbit propagation tool. Specifically, routines to model orbits involving Lagrange points and halo orbits could be implemented to further increase the level of customization. In addition, the routines and test data need to be provided to the SCENIC development team for the integration of these features into the SCENIC UI.

Appendix—Nomenclature

AP	argument of periapsis
GPS	Global Positioning System
MA	mean anomaly
NOAA	National Oceanic and Atmospheric Administration
RAAN	right ascension of ascending node
SCaN	Space Communications and Navigation
SCENIC	Strategic Center for Networking, Integration, and Communications
SMA	semimajor axis
SPICE	Spacecraft Planet Instrument C-matrix Events
TLE	two-line element
UI	user interface

Symbols

a	semimajor axis
$\mathbf{e}$	eccentricity vector
e_z	eccentricity in the z-direction
E	eccentric anomaly
f	phasing parameter
$\mathbf{h}$	angular momentum vector
h_z	angular momentum in the *z*-direction
i	orbital inclination
M	mean anomaly
n	mean motion
p	number of orbital planes
$\mathbf{r}$	position vector
r_a	apoapsis distance
r_p	periapsis distance
T	superscript in Equation (5)
t	total satellites in constellation
$\mathbf{v}$	velocity vector
w	argument of periapsis
β	ecliptic latitude
δ	declination
λ	ecliptic longitude
$\boldsymbol{\eta}$	vector pointing toward ascending node
η_x	pointing toward ascending node in *y*-direction
η_y	pointing toward ascending node in *x*-direction
μ	gravitational constant
ϕ	axial tilt
θ	true anomaly
Ω	right ascension of ascending node
α	ascension

References

1. Wiley, Larson J.; Wertz, James Richard; and D'Souza, Brian: SMAD III: Space Mission Analysis and Design, Third Ed.: Workbook. The Space Technology Library, Microcosm Press, El Segundo, CA, 2005, pp. 95–202.
2. Spacecraft Planet Instrument C-Matrix Events (SPICE) Toolkit for MATLAB. Software Package, Version N0066, Navigation and Ancillary Information Facility, Jet Propulsion Laboratory, Pasadena, CA, 2017. https://naif.jpl.nasa.gov/naif/toolkit_MATLAB.html Accessed Oct. 29, 2018.
3. Park, Ryan S.: HORIZONS Web-Interface. Jet Propulsion Laboratory Solar System Dynamics database, 2018. https://ssd.jpl.nasa.gov/horizons.cgi Accessed July 2018.
4. Standard for Models and Simulations. NASA–STD–7009A, 2013.
5. Kauderer, Amiko: Definition of Two-Line Element Set Coordinate System. National Aeronautics and Space Administration, 2011. https://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/SSOP_Help/tle_def.html Accessed July 26, 2018.
6. Kelso, T.S.: NORAD Two-Line Element Sets Current Data. CelesTrak, 2018. http://celestrak.com/NORAD/elements/ Accessed June 10, 2018.
7. Nave, Carl R.: Declination and Right Ascension. HyperPhysics. http://hyperphysics.phy-astr.gsu.edu/hbase/eclip.html Accessed Oct. 31, 2018.
8. Ecliptic Coordinate System. Wikipedia, 2018. https://en.wikipedia.org/wiki/Ecliptic_coordinate_system Accessed Oct. 31, 2018.
9. Williams, David R.: Planetary Fact Sheet. NASA Goddard Space Flight Center, 2016. https://nssdc.gsfc.nasa.gov/planetary/planetfact.html Accessed Oct. 31, 2018.