Visualization of Orbital Debris with Cesium and Satellite-js By Daniel A. O'Neil

Introduction

Analytical Graphics Inc. (AGI) provides a free open source digital globe and JavaScript Application Programming Interface (API), named Cesium. With Cesium, a developer can create interactive web-based space mission visualizations. This tutorial explains how to develop a Cesium web-app to visualize orbital debris using the Simple General Perturbations (SGP) model and Two-Line Element (TLE) data.

Prerequisites and Objectives

A TLE encodes a list of orbital elements for Earth-orbiting objects at a given point of time known as an epoch. The SGP is a set of mathematical models for calculating orbital state vectors relative to the Earth-centered inertial coordinate system. The satellite-js code library is an orbital propagator that applies the SGP to TLE to generate coordinates for web-based space mission visualizations. The Center for Space Standards & Innovation operates a web-site, named Celestrak, which provides current TLE data for satellites and orbital debris. The next section of this tutorial explains how to convert a TLE text file to a JavaScript string variable for use by satellite-js SGP orbital propagator functions. Other sections explain how to reference satellite-js and Cesium and present code listings for visualizing and propagating orbital debris.

https://en.wikipedia.org/wiki/Two-line_element_set

https://en.wikipedia.org/wiki/Simplified_perturbations_models https://github.com/shashwatak/satellite-js http://www.celestrak.com/NORAD/elements/

Data Conversion

Figure 1 depicts a TLE text file in the background and the same data reformatted as a JavaScript text string. Upon selection of a TLE data-set at Celestrak, the text file will appear in a web-page. Copy and paste the data into Excel. A simple macro can skip the rows and copy the data to another column. The following Visual Basic for Applications (VBA) code snippet copies the data without the label rows.

Celestrak.com/NOF	AD/elements/iridium-33-debris.txt 🔍 C 🛛 Q. Search	
🗿 Most Visited 🧶 Getting	Started 🔝 [Global view of Moon] 💭 Virtual AGC Home Page 🛄 Textexture	
IRIDIUM 33		
1 24946U 97051C 1	6099.52341881 .00000125 00000-0 38262-4 0 9997	
	8.6913 0006702 204.6629 155.4246 14.33479227971765	
IRIDIUM 33 DEB		
	6098.58321405 .00010025 00000-0 16200-2 0 9995	
	6.1901 0030480 183.1902 297.5142 14.70953773375032	
IRIDIUM 33 DEB	var irridiumDebris = [
2 33773 86.3985 10	"1 249460 97051C 16088.49430442 .00000105 00000-0 30870-4 0 9990 "2 24946 86.3844 113.3116 0005770 249.4421 110.6155 14.3347665897018	2",
	"1 33772U 97051K 16089.11382883 .00010997 00000-0 17867-2 0 9999 "2 33772 86.4122 120.3696 0029226 216.6976 195.8406 14.7074588437363	·.,
IRIDIUM 33 DEB		, ",
1 33775U 97051N 1	2 33773 86 3990 112 6908 0009436 300 1585 59 8678 14 3786374637341	í".
2 33/13 60.364/ 10	"1 33775U 97051N 16088.14332300 .00000509 00000-0 17385-3 0 9993	7",
IRIDIUM 33 DEB	"2 33775 86.3648 108.7377 0012236 247.6614 180.5969 14.3441903037246	3",
	"1 33776U 97051P 16089.18128044 .00000267 00000-0 89491-4 0 9990 "2 33776 86.4035 120.2769 0012531 253.0572 177.6279 14.3352411537253	
2 33776 86.4026 11	"1 33777U 97051Q 16089.10123914 .00000740 00000-0 22601-3 0 999	à"'
IRIDIUM 33 DEB	"2 33777 86.3817 106.0372 0006558 93.1508 267.0441 14.4057659337374	
1 337770 970510 1	"1 33849U 970515 16088.87583149 .00006314 00000-0 18893-2 0 9992	2",
	"2 33849 86.1074 55.1367 0070710 83.8181 277.1068 14.4129922736933 "1 33850U 97051T 16089.11929338 .00000375 00000-0 12461-3 0 999	
IRIDIUM 33 DEB	"1 33850U 97051T 16089.11929338 .00000375 00000-0 12461-3 0 999 "2 33850 86.3413 100.1894 0009713 252.4482 176.7371 14.35132113372419	
	e"1 338530 97051w 16089.15659400 .00000365 00000-0 22486-3 0 999	·,
		1".
2 33849 86.1075 5	"1 33854U 97051X 16088.84739831 .00003214 00000-0 63974-3 0 9993	3",
IRIDIUM 33 DEB	"2 33854 86.2308 40.2257 0011555 205.7788 209.2435 14.6176221537604	δ,
	"1 338550 97051Y 16089.16889664 .00009956 00000-0 27312-2 0 9994 "2 33855 86.3750 157.1567 0071972 219.9160 139.6738 14.45460079365350	÷.,
	"1 33858U 97051AB 16088.50648676 .00007911 00000-0 96609-3 0 999	5"'
	"2 33858 86.1377 8.5820 0030330 106.0560 307.1650 14.82843346376721	1",
	"1 33859U 97051AC 16088.51833561 .00001774 00000-0 56794-3 0 9998	B".
	"2 33859 86.3425 106.0653 0042957 292.8324 193.4040 14.3857040037158	3",
	"1 33860U 97051AD 16089.06011414 .00000991 00000-0 28743-3 0 9999	θ,

Figure 1 A TLE text file and the same data formatted as a String

```
Sub SkipLabels()
   Dim rgwithLabel As Range
   Dim rgnoLabel As Range
   Dim i As Integer
                            ' source row counter
                            ' destination row counter
   Dim j As Integer
   Set rgwithLabel = ThisWorkbook.Worksheets("Labeled").Range("A2")
   Set rqnoLabel = ThisWorkbook.Worksheets("noLabel").Range("A1")
   j = 1
   i = 1
  While rgwithLabel.Cells(i).Value <> ""
     If i Mod 3 <> 0 Then
       rgnoLabel.Cells(j).Value = rgwithLabel.Cells(i).Value
       j = j + 1
     End If
     i = i + 1
   Wend
End Sub
```

Additional reformatting includes adding quotes at the beginning and end of the lines, commas to separate the lines, the variable declaration, and brackets to enclose the strings. The data is saves as a JavaScript file, e.g., Irridium.js. Within the web page for the visualization, the file is included with the line, <script src="IrridiumDebris.js"></script>.

Visualization with Cesium

The Satellite-js code library implements the Simple General Perturbations (SGP) model. The **Prerequisites and Objectives** section provided links to an article about SGP and the Satellite-js GitHub project page. This section provides lines for including the Satellite-js and Cesium code libraries and code snippets for generating the points in Cesium and propagating the orbits with Satellite-js. Trajectories are propagated from TLE data provided by Celestrak. Figure 2 presents a screen-shot of the Cesium and Satellite-js demonstration. The following link leads to the demo.

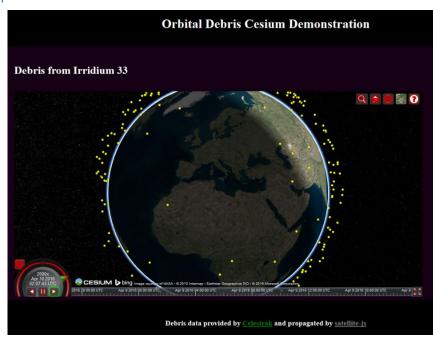


Figure 2 Screenshot of the Cesium and Satellit-js Demo

http://daoneil.github.io/spacemission/Apps/Cesium_with_SGP.html

Referencing the Code Libraries

The previous section provided a line to include a JavaScript file that defines a variable that contains the TLE data. The following lines include the scripts for Cesium and Satellite-js.

```
<script src="../Build/Cesium/Cesium.js"></script>
<link rel="stylesheet" type="text/css" href="satellite-js-
master/sgp4_verification/css/app.css">
<script src="satellite-js-master/dist/satellite.js"></script>
<script src="satellite-js-master/dist/satellite.js"></script>
<script src="satellite-js-master/dist/satellite.min.js"></script>
<script src="satellite-js-master/dist/satellite.min.js"></script>
<script src="satellite-js-master/dist/satellite.min.js"></script>
<script src="satellite-js-master/dist/satellite.min.js"></script>
</script src="satellite-js-master/sgp4_verification/lib/angular/angular.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></scrip
```

JavaScript Code for Cesium

The following code snippet loops through each TLE and adds a point entity to an array. A loop counter becomes the id for the point and an index into the array.

JavaScript Code for Satellite-js

Functions within the Satellite-js library read the TLE and create a record, propagate position based on advancing time, and generate Cartesian coordinates in an Earth Centered Fixed or Earth Centered Inertial reference frame. The following code snippet generates an array of debris records from the TLE data, generates an array of positions and velocities, and propagates the position and velocity array.

```
// Declare orbital debris variables
var debrisRecords = [];
var datasetSize = irridiumDebris.length ;
                             // positions and velocities of orbital debris
var posVel = [] ;
function propagateOrbitalDebris() {
  var j = 0;
  for (i=0; i < datasetSize; i++) {</pre>
    var tle1 = irridiumDebris[j] ;
    var tle2 = irridiumDebris[j + 1] ;
    if (typeof tle1 == 'string' || tle1 instanceof String || typeof tle2 ==
        'string' || tle2 instanceof String) {
        debrisRecords[i] = satellite.twoline2satrec(tle1, tle2) ; }
        j = j + 2; // advanced to the next TLE in the array }
// Propagate debris using time since epoch
   for (i=0; i < datasetSize; i++) {</pre>
      if (debrisRecords[i] != undefined) { posVel[i] = satellite.sqp4
         (debrisRecords[i], timeSinceTleEpochMinutes); }
   }
```

```
// Propagate debris using time since epoch
for (i=0; i < datasetSize; i++) {
    if (debrisRecords[i] != undefined) {
        posVel[i] = satellite.propagate(
            debrisRecords[i],
            now.getUTCFullYear(),
            now.getUTCMonth() + 1, // Note, function requires the range 1-12.
            now.getUTCDate(),
            now.getUTCDate(),
            now.getUTCHours(),
            now.getUTCMinutes(),
            now.getUTCSeconds()
        );
        } //endif
}//next i
}//end propateOrbitalDebris</pre>
```

Demonstration Source Code

A GitHub repository, <u>https://github.com/daoneil/spacemission</u>, includes the files named Cesium_with_SGP.html and IrridiumDebris.js. The HTML file is the web-page with embedded JavaScript that calls functions in the Cesium and Satellit-js libraries. The JavaScript file defines the variable irridiumDebris, which contains the TLE data for the debris from Irridium 33.

Conclusions

This tutorial provided links to information about SGP, Celestrak, TLE, and Satellit-js. A data conversion section explained to use Excel to format the data as a JavaScript string variable and it include VBA code to copy the data without the headers. A visualization section provided code snippets for referencing the code libraries, creating the point entities that represent the data, and propagating positions based on time advancement. A source code section provides the link to the GitHub repository and identifies the source code files. Potential follow-on development efforts include adding additional debris data sets, changing the point entities to 3D objects, and creating a maneuverable spacecraft for capturing the debris objects.