MAE3145: PREDICT

Description

You are tasked to support the elite Space Surveillance Reconstitution Team (SSRT) as an orbital analyst. The SSRT will deploy to setup a remote space tracking capability when threat levels against the US indicate an attack is imminent. As a SSRT member, it is your responsibility to develop software to locate satellites form an Earth location. In particular, SSRTs tactical terminal requires topocentric range, azimuth, and elevation to point its new secret tracking instrument. Your program must compute and output this information at <u>two minute intervals</u> each time the tactical <u>terminal is in darkness</u> and the <u>satellite is above the terminals local horizon at a reasonable range</u>. In addition, your program must determine if the satellite is visible to the tactical terminal, i.e. terminal in the dark, satellite illuminated by the sun, elevation angle greater than or equal to 10°, range less than 1500 km. If the satellite is in sunlight, SSRT members can operate the terminal's new pointing instrument in one of three classified modes.

The SSRTs existence must be kept secret. Thus, to validate your software **you must** view a low Earth orbiting satellite based on your program results. In addition, if you know someone located at least 100 km from our training site, and this person can be sworn to secrecy, give them your program results so they too can view a satellite. Another person viewing a satellite based on yoru results will serve as a independent validation of your work.

As with all practical programming applications, there are three steps to this project:

1. You will develop the correct algorithm and accomplish some hand calculations to provide a basis to test your coded subroutines,
2. You will code all subroutines, checking their output with that of your hand calculations to validate each subroutine as you go, eventually putting together the complete program and matching a provided single satellite data file perfectly,
3. You will slightly alter this working program to accept a data file of hundred of real satellites, identify a viable vehicle to personally view, and go view it.

The data required to run your program is located in the MAE3145: Astro Library. The data comes from JSPOC's Two-Line Element Sets (TLEs), which are typically found in terms of three lines of data. Each data set contains the 100+ brightest active payloads that JSPOC tracks. However, not all satellites will be visible to the naked eye.

The TLE files are given in a very specific format as shown in Fig. 1. The following describes each parameter contained in the TLEs. Some of these numbers will not be used in your program. You may also get the latest TLEs yourself at celestrak.com. There are many references for the TLE description available in textbooks or online, but one is available https://goo.gl/W6MZb2.

Project Requirements

After completing the project you must submit the hard copies of your work:

- Complete algorithms for the main driver script as well as a seperate algorithm for each sub-function that you develop. Someone totally unfamiliar with astrodynamics should be able to duplicate your program in any computer language.
- Clear, concise and properly documented and tested code

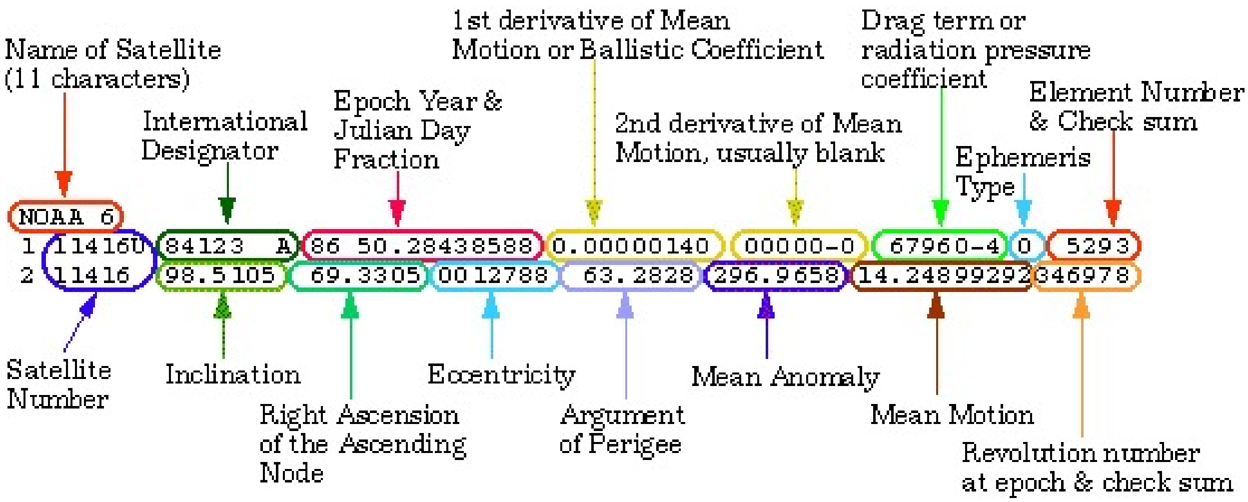

Figure 1: TLE Format

- Correct outputs from your program which matches the test cases
- Any additional test cases you may have used. Explain why you did or did not use any additional test cases.

Authorized Resources

You may consult with your instructor, the course notes or other reference material, and other students. However, you **MAY NOT** copy another student or any other individuals code. The program you develop must be your own work.

Algorithm

Write a structured algorithm that shows your approach to writing a computer program to perform all of the tasks described above. This should be a complete **sequential** list of the **equations and logic (including loop)** that you will use to write your program. The details of subalgorithms are only required for procedures that are new for this project, not those provided to you. Instead, just mention what procedure will be used and the inputs and outputs of the procedure, e.g. "Calculate orbital elements given position and velocity vectors using `rv2coe`".

When completed, anyone should be able to write your code **solely** using the algorithm in **any** computer language of their choice. Thus, define all symbols before you use them, and do not write the equations and logic using any language specific terminology, i.e. Something like " Find the length of the vector using `norm(x)` " is unacceptable.

Your algorithm must be **typed**, which will serve you well when you document your final code. This is also a good opportunity to practice your technical writing skills in LaTeX.

Final PREDICT Deliverables

Your program must process the `PREDICT.DAT` data file and generate output that matches the orbital elements in `PREDICT.OUT` to at least four decimal places.

Submit the following on the due date:

- Fully documented driver script

- Each procedure which you wrote or modified, fully documented (no library routines)
- Computer generated results which match `PREDICT.OUT`

Program Specifications

The following is a description of the required functions that your softwre program must include. You will be making use of many of the functions that you have been developing over the course of the semester.

`J2DragPert`

This procedure calculates the rates of change of the right ascension of the ascending node, argument of periapsis, and eccentricity. The perturbations considered are limited to J_2 and drag only.
Inputs:

- i_0 – initial inclination in radians
- e_0 – initial eccentricity
- n_0 – initial mean motion in radians per second
- $\frac{\dot{n}}{2}$ – mean motion rate divided by two in radians per second squared

Output:

- $\dot{\Omega}$ – nodal rate in radians per second
- $\dot{\omega}$ – argument of periapsis rate in radians per second
- $\dot{e}$ – eccentricity rate

Update

This procedure uses the method of general perturbations to update the classical elements from t_0 to $t_1 = t_0 + \delta t$ for inclined elliptical orbits. It includes the effects due to first-order secular rates (second order for mean anomaly) caused by drag and Earth oblateness (J_2). It will be necessary to call `newton` from this function to calculate the future value of eccentric anomaly and true anomaly.
Inputs:

- δt – elapsed time since t_0 in seconds
- n_0 – mean motion at t_0
- $\frac{\dot{n}}{2}$ – mean motion rate divided by 2
- e_0 – eccentricity at t_0
- $\dot{e}$ – eccentricity rate
- Ω_0 – right ascension of the ascending node at t_0
- $\dot{\Omega}_0$ – right ascension of the ascending node rate
- ω_0 – argument of periapsis at t_0

- $\dot{\omega}$ – argument of periapsis rate
- M_0 – mean anomaly at t_0

Outputs:

- n – mean motion at t_1
- e – eccentricity at t_1
- Ω – right ascension of the ascending node at t_1
- ω – argument of periapsis at t_1
- ν – true anomaly at t_1

coe2rv

This function computes the spacecrafts position and velocity in the Earth centered inertial frame from the classical orbital elements.
Inputs:

- n – mean motion at t_1
- e – eccentricity at t_1
- Ω – right ascension of the ascending node at t_1
- ω – argument of periapsis at t_1
- ν – true anomaly at t_1

Outputs:

- r – position vector in ECI
- v – velocity vector in ECI

visible

This function will determine if the observer is in the dark. If so, it will then compute the topocentric range, azimuth, and elevation of the satellite relative to the observer. If the elevation is greater than $10°$ and the range is less than 1500 km then the function will check if the satellite is illuminated by the sun. If the satellite is visible then a flag will be set to true.
Inputs:

- r – position of the satellite in the ECI frame
- r_s – position of the observer in the ECI frame
- λ – observer geodetic latitude
- θ – observer local sidereal time
- JD – Julian date at viewing time

Optional Outputs:

- ρ – range
- α – azimuth
- ϵ – elevation
- d – flag for observer darkness
- v – flag for visibility of satellite

These outputs are all optional. You may decide to use this function to print your output data, or pass data to another function to output.

rhoazel

This function determines the topocentric range, azimuth, and elevation from the observation site to the satellite.
Inputs:

- r – position of satellite in ECI frame
- r_s – position of observer in ECI frame
- λ – observer geodetic latitude
- θ – site local sidereal time

Output:

- ρ – range
- α – azimuth
- ϵ – elevation

PREDICT Final Report

Your program must process a TLE file and generate topocentric data so you can view a satellite. You are required to summarize your project results and submit and engineering report (not to exceed 5 typed pages, excluding title page, table of contents and appendices) to your instructor. Your report should contain the following sections:

1. **Intro and Assumptions** – Introduce and motivate the problem, then briefly discuss the assumptions that were required to solve the problem. You may simply list, without discussions, any assumptions being used that have been used in previous projects.
2. **Math Technique** – This section is a summary of the theory that we used to solve the problem, with an emphasis on the math that is new since `COMFIX`. Be sure to address why each step was taken. Do not discuss your computer code here, i.e. " Next I wrote `rsite`…" and don't simply refer to or regurgitate your algorithm.

3. **Analysis/Discussion** – Introduce your code here and include a discussion of how it was validated, including options for validating. Discuss the output you generated, including a summary of the data on the satellite, or satellites you viewed, along with details of that viewing as compared to what was expected.

4. **Sources of Error/Recommendations** – Discuss any errors, that is, where your assumptions impacted your results. Include any recommendations for how to improve the results.

5. **Appendix A – Program Usage** – You must include instructions so a person can you your executable code to generate observation data for any site, range of days, and any number of TLEs. Write this section of your report as **if it were a page in a user's manual**. Begin by briefly saying what your program does. Describe how someone would use your program. Are there any limits, e.g. matrix size or range of values? Is there anything unique to your program? Are there any data for which teh program will not work? As an example here's a good format you can use:

   - Program Name
   - Program function
   - Computer used/limits
   - Language and version
   - How to run it
   - Input
     - What are your inputs/units/limits?
     - What method is used to input the data, i.e. how do you change the input data?
     - Point out user interactive prompts if there are any.
   - Output
     - Explain the method used to output the data?
     - What are your outputs/units?

6. **Appendix B – Results** – Program results for at least 3 objects from the TLEs (**NOT ALL 100+ please!**), over the time interval of your choice, to include when you observed a satellite.

7. **Appendix C – Bonus** – Pick a location more than 100 km from DC and run your program. If you or any other person view a satellite based on your computer results, you will receive a bonus on your project