Manual Orbit Interpolation

Aymeric Labarbe, Anna Schwab, Philippe Gigon
SSA EPFL Team

May 20, 2022

1 Input format

The input consists in three IODs[1] of the following format in the file "`IOD.txt`":

00000 03 123XYZ 0000 G YYYYMMDDHHMMSSsss 00 45 DDDMMSSs+DDMMSSs 00 B+MMm Mm SSSsss

With YYYYMMDDHHMMSSsss, giving the year, month, day, hour, second and milliseconds of the observations (UTC Time), DDDMMSSs+DDMMSSs gives the azimuth and elevation angles of the satellite with D degrees, M minutes (i.e. 1/60°) and S seconds (1/3600°) The position of the observer must be given given in the file "`LOCATION.txt`".

2 Gauss algorithm

The main part of the orbit interpolation is the Gauss algorithm which follows the algorithm provided in Orbital Mechanics for engineering students by Howard Curtis [2]. The algorithm makes a first guess of the position and velocity at the time of the second observation and then itteratively improves the result. The final output of the algorithm are the velocity and position of the satellite at the second observation time. The algorithm supposes that the three provided angles are close to each other without specifying what this means. In practice the algorithm is a bit capricious in the sense that the results can be everything from nearly perfect to completely absurd. This is not due to the implementation which we tested carefully with textbook examples and other data, but is rather an intrinsic problem of the Gauss algorithm that is described by many different authors. Finally, note that the best results are obtained for satellites at 60° elevation. Observations close to the horizon always give absurd results.

3 Output and TLE

The ultimate goal of the algorithm is to find the TLE (Two Line Element) of the satellite, which is the output format of the code. An example is shown in figure 1. In the first line, the only important value for this project is the epoch time, when the observation of the satellite was done. The other elements of the first line correspond to administrative data of the satellite.

The second line of a TLE is dedicated to the orbital elements. The only exception is the very first element, which corresponds to the NORAD satellite catalogue number. It is set to 00000 in the expample in figure

[1]For more details check: `http://www.satobs.org/position/IODformat.html`

[2]Howard Curtis, Orbital Mechanics for Engineering Students, p.235-242, ISBN 0 7506 6169 0

```
1 00000U 00000000 22117.44211806 0.0000000 0000000000 00000000 0 00000
2 00000 097.2225 168.7478 0016232  84.9179 337.6234 15.367583120000000
```

Figure 1: Two line element

1. The following elements are: inclination i, right ascension of the ascending node Ω, the decimal places of the eccentricity ϵ, argument of perigee ω, mean anomaly M, the mean motion n followed by the relevation number and the checksum, which have been set to 0 in this project as well. The orbital elements are all calculated in the file "`TLE_class.py`", where also the format of the singel elements is defined. Most of the formulas used to calculate the orbital elements are taken from the already cited book Orbital mechanics for engineering students by Howard Curtis.

4 Obtaining the orbital elements

In order to obtain the position, velocity and the orbital elements, one needs to first create the IOD in "main.py". Then one can choose convergence parameters for the iterative Gauss algorithm. Calling "find_r" will return the position and the velocity at time t_2 which then allows to compute the TLEs. In general it is not necessary to modify anything except the input files. With the found position and velocity and with the time t_2, an object `TLE` can be created. A simple list of the orbital elements in the order [ϵ, n, i, Ω, ω, M, t_2] can be created with the function `list_TLE` in the class `TLE`, which allows to check the calculations. The complete TLE in the right format can be created with the function `TLE_format` also in the same class. A separate file is created called `TLE.out`.

5 Image Treatment

The code already written partially by Yann Bouquet for the project *Delineating Satellite Tracks in Astronomical Images*, and by Manon Béchaz, Baptiste Claudon, Jules Eschbach and Salomon Guinchard for the project *Detecting orbital objects with optical camera*, has been improved in the folder "Several images". The updated version works as the old one, one should consult the `README`-file in the folder. The only change in the input format is that the times of all the pictures in the folder should be added to the input file `parametres.txt`, such as it is done in the example folder `Sat1`.
A modified version has also been created, which is able to extract the three IODs need for the computation of the position and velocity from only one image, meaning one satellite trace. This version is saved in the folder "One image", including a folder "TestOne", where an example is given of how the code has to be used. It follows closely the `README`-file mentioned before. The algorithm creates an IOD file with the three points beginning, middle and end of the trace.

6 Errors

The following errors can occur during the execution of the code:

- "There must be three IOD's in ", if there are not three IODs in the file "`IOD.txt`".
- "IODs have wrong format", if the IODs are not corresponding to the model format given above. Note that all the unknown parts must be filled out, i.e. if the precision of the angles is less than given in the model IOD, one must fill up with zeros in order to exactly respect the format.
- "No root found" and "Multiple roots found" are errors related to the first estimation of the velocity and position vector. The algorithm includes the solution of a polynomial of degree 8, hence it might be that multiple roots are found. In this case one needs to check the two solutions manually and choose the better one. For all our test cases this error never appeared.
- "Eccentricity is greater than 1", this error might appear if the orbit interpolation fails completely. If the obtained results by the Gauss algorithm are very imprecise, it might be that the predicted orbit is not closed and hence has an eccentricity larger than one.

The several errors that can occur during the image treatment are not listed here. Instead, one should consult the two reports of the original projects, where the code was created.

7 Needed improvements

The implemented algorithm is only useful for preliminary orbit determination. This means that the results will never be very precise and even in the best working cases, the orbits may be off by many kilometers. Having that said, the use of the algorithm is to obtain a first idea of the orbits. This allows to detect the same object at a later pass and to improve the measurements. For this it is necessary to go beyond preliminary orbit determination and to implement statistical improvement of the orbit. The Gauss algorithm itself is very limited and can only be improved slightly, the following two improvements could be useful in the future:

- Find a way to pass from geocentric observations to topocentric coordinates. This would allow to find the angles using platesolving and hence with higher precision. As for now, it seems that this coordinate conversion is very difficult for satellites orbiting earth.
- The Gauss algorithm uses a first order approximation to the Lagrange coefficients. One could try to go beyond this approximation as done by different authors.
- The calculation of the epoch time does not take into account the leap years. In order to complete the algorithm, this should be definitely involved.
- The eccentricity calculations are unfortunately very imprecise. This phenomenon has been observed in other projects kas well and should therefore not be caused by a bug in our code. One way of improving the results is increasing the precision of the input values (angles of the observation in the IODs). After a discussion with experts, the optimal solution seems to be the following: Since the LEO satellites (Low Earth Orbit) have a nearly circular orbit, which corresponds to an eccentricity of 0. This value should be imposed and used to predict the satellite's position at a later time. Based on this prediction, more observation of the same object should be done and the the error between the prediction and the actual observation should be analysed and ultimately minimized, as a function of the eccentricity.
- The success of the image treating does depend strongly on the background noise, which is unfortunately very high for short exposition times. This is a problem that is being treated at the moment (May 2022). Once this is solved, it should be tested, whether the existing code can be adapted, or otherwise the new one inserted. The ultimate goal should be the connection of the image treatment and the orbit interpolation.