Connectonomicon

H. A. Güler

December 2024

1 Background (skip this, just trivia)

Abstract

This is not a report, I just find the format convenient.

Connectonomicon is a tool that was originally intended to be an automated tracklet identification tool for minor planets. The inspiration partly came from a couple of citizen science projects which generated asteroid tracklets, and I have been trying to keep track of some of those tracklets since. Then I decided to do something about it myself and develop some software.

While in development, I realized that I would probably never be able to reduce the errors low enough for short tracklets to allow for the proper use of automation (unless I did some Monte Carlo'ing), so instead I shifted my focus on a human-aided version of the tool with minimal minimum interaction to run.

I also realized that such a tool might potentially help some other astronomers out there. I am aware that fully automated tracklet identifications of minor planets can be done, and ARE done, but I haven't had access to such tools, and there might be some out there who likes pretty graphs.

Perhaps one day this may get developed into an automated software, if someone (likely me) spends more time on it.

2 The Workings of Connectonomicon (read this part instead)

The program internally uses km as distance units, s for time units, deg as angle units. I am currently questioning the use of degrees as it results in very tiny numbers, and working with very tiny and very large numbers all at once can cause numerical errors here and there.

2.1 Orbit Determination

2.1.1 The Built-In (Native) Method

The built-in orbit determination is half based on the Herget's method as described in Bill Gray's Project Pluto website [1]. However, to use the method, one needs to assume some initial distance to the object since a Right Ascension - Declination observation locates an object in 2 dimensions, which is one less than the 3 required to locate an object within the Solar System which happens to reside in (at least locally) 3-dimensional space. There are at least two methods that I know of that employ some sort of distance estimation, and I don't think I would be wrong to say that they are classics: method of Gauss and method of Laplace. The latter is somewhat described in [2].

Connectonomicon follows the method of Laplace until an initial distance assumption is found, then locates the object in Heliocentric space by using the known RA-DEC astrometry and the assumed distance. After that, the initial velocity is initially assumed to be the one that belongs to a circular orbit with least inclination.

$$\boldsymbol{v_0} = \sqrt{\frac{\mu}{\rho_0}} \cdot \frac{-p_{0y}\hat{\boldsymbol{\imath}} + p_{0x}\hat{\boldsymbol{\jmath}}}{\rho_0} \tag{1}$$

...where μ is the standard gravitational parameter of the Sun and ρ is the Heliocentric distance. The circularity is obtained by the use of the $\sqrt{\frac{\mu}{\rho_0}}$ velocity magnitude, and the least-inclination is obtained by zeroing the p_{0z} component.

The program then starts an iterative refinement process, nudging the initial position and velocity components in directions which reduce the sum-of-errors in RA and DEC belonging to all known observations of the object. This is similar to numerical differentiation where the derivatives of the error relative to the initial position

$$\frac{\partial \Sigma \sigma_{RA-DEC}}{\partial \boldsymbol{p_0}} \tag{2}$$

and initial velocity

$$\frac{\partial \Sigma \sigma_{RA-DEC}}{\partial \boldsymbol{v_0}} \tag{3}$$

are calculated. The state vector is then "nudged" towards the direction of adjustment where the reduction of error $\Sigma\sigma_{RA-DEC}$ is greatest. The error for a particular observation is calculated as:

$$\sigma_{RA-DEC} = (RA_{obs} - RA_{pred})^2 + (DEC_{obs} - DEC_{pred})^2 \tag{4}$$

Square rooting this equation seemed unnecessary to me as it was to be used for a comparison of scalars. The directions of nudging that are tested by the program in each iteration are in the principal cartesian directions x, y and z in ecliptic Heliocentric coordinate system.

The program stops after a number of iterations are reached or the error belonging to the final observation is reduced below some arcseconds. (In hindsight, maybe the least squares should have been the tolerance metric.)

2.1.2 Using find_orb

If find_orb is to be used, the program uses the known observations file as an argument for the fo64 executable. The fo64 executable generates an elements.txt file, which is in the following format:

```
Orbital elements:  H259629
   Perihelion 2019 Aug 31.04254 +/- 357 TT =  1:01:15 (JD 2458726.54254)
Epoch 2019 Apr  6.0 TT = JDT 2458579.5    Ju: 0.6594              Find_Orb
M 336.25908731 +/- 60                  (J2000 ecliptic)
n   0.16145607 +/- 0.0791              Peri.  100.27936 +/- 80
a   3.34015367 +/- 1.09                Node   143.57894 +/- 16
e   0.3385933 +/- 0.146                Incl.    5.23079 +/- 4.3
P   6.10449                 H 20.19  G  0.15   U 11.1  SR
q 2.20919999 +/- 0.441     Q 4.47110735 +/- 1.46
From 6 observations 2019 Apr. 6 (3.9 hr); mean residual 0".11
# State vector (heliocentric equatorial J2000):
#    -2.316087940379  -0.664911245376  -0.098921463954 AU
#    +5.774280493782 -10.384539669698  -3.953074747089 mAU/day
# MOIDs: Me  1.749729 Ve  1.487210 Ea  1.203666 Ma  0.693187
# MOIDs: Ju  0.659429 Sa  4.612909 Ur 14.704471 Ne 25.362050
# 1 oppositions
# Elements written: 14 Dec 2024 23:18:08 (JD 2460659.470930)
# Full range of obs: 2019 Apr. 6 (3.9 hr) (6 observations)
# Find_Orb ver: 2024 Oct 20
# Perturbers: 00000001 ;  JPL DE-405
# Tisserand relative to Jupiter: 3.05928
# Diameter 377.0 meters (assuming 10% albedo)
# Score: 0.445503
#  $Name=H259629  $Ty=2019  $Tm=08  $Td=31.042544  $MA=336.25909
#  $ecc=0.3385933  $Eqnx=2000.
#  $a=3.3401537  $Peri=100.27937  $Node=143.57895  $Incl=5.23080
#  $EpJD=2458579.500  $q=2.209200  $T=2458726.542545  $H=20.2
# Sigmas avail: 3
```

The program reads this file and uses the state vector lines to place the object on an initial position in ecliptic Heliocentric space.

2.2 Orbit Propagation

The orbit propagator uses SPICE kernel DE440 to obtain the planetary barycenter positions at any time. It then computes the gravitational acceleration on the minor planet as a sum of Newtonian gravitational accelerations due to all bodies; the Sun and the planetary barycenters. The time-step sizes are allowed to be somewhat large, relying on the propagator errors not growing large enough to match the orders of magnitude of errors due to initial state vector errors for short arc observations. Needless to say, this will fail to capture the motion around large-gravity-gradient locations such as close approaches to any of the planets, eg. the Earth-Moon system.

The propagator solver is an 8th order leapfrog method proposed by Haruo Yoshida, [3] often referred to as Yoshida method.

2.3 Observation Classifications

Potential observations are classified according to their times, locations and magnitudes.

- The dates of potential observations must be within the orbit propagation time or just a little ahead (eg. by 20% of the propagation time)
- The sky positions must be closer to the path traced by the propagated orbit than some tolerance (eg. 20 arcseconds)
- The magnitude should not vary more than a given value (eg. 2)

If all conditions match for a tracklet, it is marked as a potential match. Still, unmatched observations can be displayed on the resulting chart.

2.4 Linear Approximation

For short propagation times (eg. not more than few days for a MBA), it is usually okay to approximate minor planet's sky coordinate motions as constant-speed straight-line motions. This is also employed in Connectonomicon.

In addition, one can give an estimation of their maximum errors in astrometric measurements related to the images on which the measurements were made. If the measurement was a pixel off from a would-be true-to-real-life measurement, the slope of the line of motion $\Delta DEC/\Delta RA$ could be off by some amount related to the angular resolution of the pixels. The speed could also be off by some amount. Using a max and min slope as well as a max and min speed estimate, an area of predicted locations can be generated. This is done by taking 4 points along two observations to be examined, each of the 4 points off from the measured location by the given amount of error (in pixels) in positive and negative RA and DEC directions.

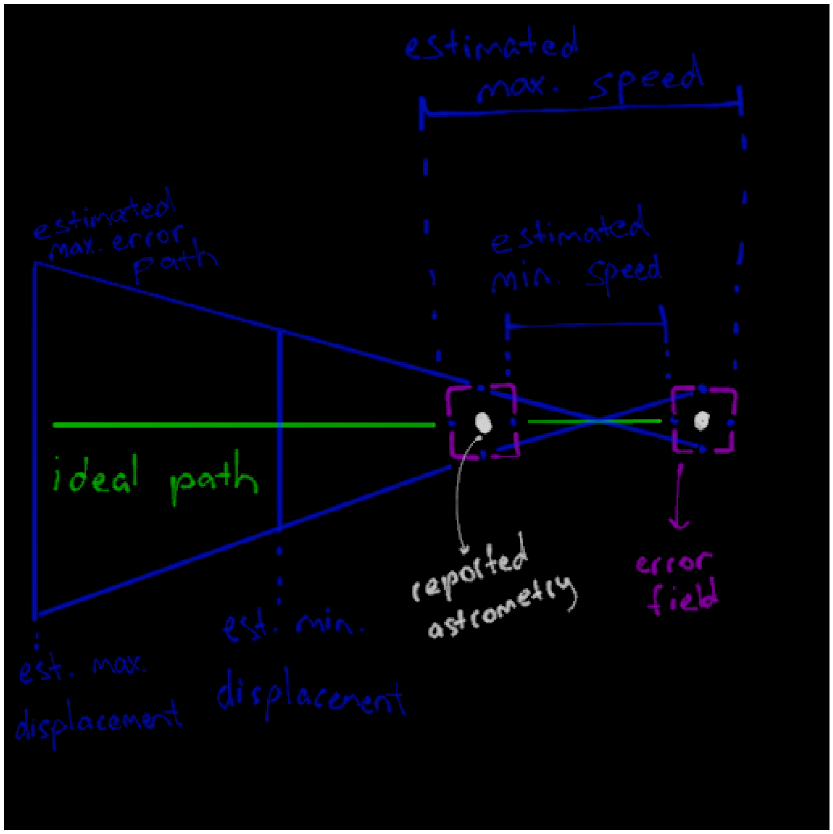

Figure 1: Visualization of astrometry reporting errors

References

[1] B. Gray. Method of herget: mathematical details. `https://www.projectpluto.com/herget.htm`. Accessed: 2024-12-14.

[2] L. G. Taff. The resurrection of laplace's method of initial orbit determination. Technical report, MIT Lincoln Laboratory, 1983.

[3] H. Yoshida. Construction of higher order symplectic integrators. *Physics Letters A*, 150(5):262–268, 1990.