Update to Bernstein-Khushalani Orbit-Fitting Software

G. Bernstein, Date : 2001/09/1419 : 05 : 06

The orbit-fitting software uses the methods described by Bernstein & Khushalani 2000, "Orbit Fitting and Uncertainties for Kuiper Belt Objects," *AJ* **120** 3323. I have since refined the algorithms used by the software to better recognize and fit degenerate orbits, and to incorporate features such as orbiting observatories. This document describes the updated algorithms, and describes the other conveniences and features added to the software since its original release to the public. I assume that you've read the original paper before reading this document. The `README` file with the software is a more complete guide to installing and running the software.

Please remember that the software calculates **barycentric** *osculating elements, not the more commonly used heliocentric elements.* The barycentric elements are more stable for KBOs, but disagree with the MPC's conventions.

1 Algorithms

1.1 Recognizing and Fitting Degenerate Orbits

In the original code, the steps for recognizing and fitting degenerate orbits were as follows:

1. Do a preliminary linearized 5-parameter fit to the observations, with $\dot{\gamma}$ set to zero. This step was skipped if the number of observations is only 2, since the fit would be degenerate.

2. Use the linear fit as a starting point for a full non-linear 6-parameter fit.

3. If the 6-parameter fit has a large $\mathrm{Var}(\dot{\gamma})$, then set $\dot{\gamma} = 0$ and execute a non-linear fit to the other 5 parameters, placing a nominal value for $\mathrm{Var}(\dot{\gamma})$ into the output covariance matrix Σ.

4. If the 5-parameter fit still has unbound orbits within the uncertainty range, fit again but add an energy constraint to the χ^2 being minimized, reducing the degrees of freedom by one.

I have realized that Step (3) of this process can lead in some cases to orbit fits that are misleadingly precise. This occurs when the observations are very accurate in constraining 5 degrees of freedom, but still have a strong degeneracy. Examples are: having only two observation nights but with several months between them; HST observations, with highly precise astrometry but over a short arc (few days); or two or more observations, but all at nearly the same time of year, so that line-of-sight motion remains indeterminate.

In these cases the degeneracy is primarily, *but not exactly,* along $\dot\gamma$. Then when we set $\dot\gamma = 0$, we force the solution to a very precise location in the six-dimensional space and appear to get a very precise orbit, when in fact it is strongly degenerate.

To remedy this situation, and to more clearly recognize when the orbit is doubly degenerate, the procedure is now as follows:

1. Do a preliminary linearized 5-parameter fit to the observations, with $\dot\gamma$ set to zero. The uncertainty on γ, *i.e.* the KBO distance, is examined, and if

$$\frac{\sigma_\gamma}{\gamma} > 0.2 \tag{1}$$

   then we immediately recognize that the arc is too short to constrain even 5 parameters, and the orbit is doubly degenerate.

2. If we pass the above test, then we execute the full 6-dimensional fit, as before.

3. If the 6-parameter fit has $\mathrm{Var}(\dot\gamma)$ comparable to the velocity needed to unbind the orbit, then we recognize the orbit as singly degenerate. In this case, the 6-parameter fit is repeated, but with a constraint on energy so that we are minimizing a pseudo-χ^2

$$\tilde\chi^2 \equiv \sum_i \left[\frac{(\hat\theta_{x,i} - \theta_{x,i})^2}{\sigma_i^2} + \frac{(\hat\theta_{y,i} - \theta_{y,i})^2}{\sigma_i^2} \right] + \frac{f_b^2}{3}, \tag{2}$$

   where the binding energy fraction f_b is defined by

$$\dot\alpha^2 + \dot\beta^2 + \dot\gamma^2 = (1 + f_b) G M_\odot \gamma^3. \tag{3}$$

   Again, $f_b = 0$ corresponds to a circular orbit, and $-1 < f_b < 1$ for bound orbits. Hence the addition of the energy constraint pushes the orbit to be bound, and close to circular.

4. If there were only 2 observations, or if Step (1) indicated a doubly degenerate orbit, then the $\tilde\chi^2$ minimization is done with $\dot\gamma$ fixed at zero, with a nominal variance placed on $\dot\gamma$ post-facto.

Hence the primary change is that we use the energy constraint, rather than setting $\dot\gamma = 0$, as a means of handling singly-degenerate orbits. This avoids the (sometimes erroneous) assumption that *all* of the degeneracy is in $\dot\gamma$.

Another bug which was found to arise is that the traditional means of minimizing χ^2 and calculating Σ is to ignore the second derivatives of the fitted quantities, as is done in the standard *Numerical Recipes* routines that I used. This approximation turns out to often be a poor one for the f_b term in our $\tilde{\chi}^2$. The code now explicitly inserts the necessary second-derivative terms. This bug had caused unrealistically small error bars to be placed on some very degenerate orbits.

The default error placed on $\dot{\gamma}$ for doubly-degenerate orbits, and the strength of the constraint on f_b, are weaker than in the last release of the software, meaning the degenerate orbits will show somewhat larger errors in position and elements than before.

1.2 Orbiting Observatories

In preparing a proposal for KBO observations with HST, I produced code to compute positions from observatories in orbit around the Earth. If an observation uses an observatory code which is ≥ 2000, it is assumed to be an orbiting observatory. The `observatories.dat` file must containing a corresponding entry with the following data on a line:
obscode inclination period precession_period jd0 RA0 name
with the inclination (degrees) and period (days) of the spacecraft orbit, the precession period (in days) of the orbit pole about the Earth pole, the JD at which the spacecraft is at the ascending node of the orbit with pole directed toward a particular RA. The supplied `observatories.dat` file contains an entry which allows simulation of the HST orbit. This is a schematic representation of the orbit, of course, just simulating the motions and seeing the effect on orbital fits, parallax, etc. Precise actual data must know the phase of the HST orbit more carefully than this simple parameterization can provide. Indeed due to variable drag it is impossible to predict the HST position to better than a few thousand km even a month in advance.

The spacecraft orbit is taken to be circular.

2 New Conveniences

The basic programs now accept command-line arguments through which one can specify the location of the binary JPL ephemeris and/or the observatories database file. Also the uncertainty assigned to observations in MPC format can be altered on the command line.

Alternatively the locations of the ephemeris and observatory files can be specified by setting the environment variables `ORBIT_EPHEMERIS` and `ORBIT_OBSERVATORIES`. For example your `.cshrc` file could contain the line
`setenv ORBIT_OBSERVATORIES /usr/local/orbits/observatories.dat`
If neither of these is done, the programs look in the current directory as before.

The Release code for the software can be found by putting the `-v` option on the command line of most of the basic programs as well.

Sample input and output files for Pluto and 1992 QB_1 are now included in the distribution, to help check for proper compilation.

3 New Subroutines

The `ephem_earth.c` code has been reorganized to handle the orbiting observatories described above. Some additional subroutines have been created, however, which are useful in creating your own programs:

- `elements_to_xv()` will create a Cartesian-basis orbit representation from an orbital-element representation; previously, only the reverse transformation was available.
- `zenith_angle()` will calculate the angle between an observation direction and the anti-geocenter vector.
- `zenith_horizon()` will calculate the angle from zenith to horizon for a chosen observatory. A spherical Earth is assumed for orbiting observatories, while ground-based observatories just return 90 degrees.
- `is_visible()` tells whether a given observation is above the horizon.

Other changes to the code since the first release include the addition of more observatories to the included `observatories.dat` file. I believe the file currently contains all sites that have contributed KBO observations at $R > 21$ to the MPC.

The package is now under CVS control, so each file has a header that gives the version number. If you find bugs, etc., please let me know the version information of the files containing the bugs.