OpenOrb

Mikael Granvik[1,2] & Lauri Siltala[1]

[1] U Helsinki, Finland / [2] Luleå U Tech, Sweden

What is OpenOrb?

- Orbit computation software (primarily aimed for analyzing data of real objects).
- Development initiated at U Helsinki in 2000 by MG as a more modern and flexible alternative to the orbit-computation software used back then.
- Publicly accessible open-source sw (GPLv3) since 2010 as a response to the needs of LSST.
- Currently available through GitHub (https://github.com/oorb/oorb) with 36 forks and about 5–6 active contributors.
- Programmed primarily in Fortran 90/95 using an object-oriented programming paradigm, but (limited) access through Python also available.
- Source files contain >120k lines of code or documentation.
- Results in agreement with those by JPL and SpaceDyS.

What can one do with OpenOrb?

- Orbit computation from the discovery moment onward with particular focus on rigorous uncertainty estimation (statistical ranging, virtual-observation MCMC, differential correction, etc).
- Conversion of orbital-element sets from one type of parameters to another type.
- Assessment of short-term impact probability with planets, the Moon, and the Sun.
- Classification of objects based on their orbital-element probability-density functions (e.g., NEO/MBO/TNO).
- Rigorous ephemeris prediction.
- Propagation of orbital-element pdf.
- Observation planning (given user-defined constraints, when is an object observable from a given observatory).
- Linking astrometric observations within apparitions (short term) and across apparitions (long term).
- Asteroid mass estimation based on close encounters (fit simultaneously for orbital elements and masses for multiple asteroids).
- …

Development philosophy

- OpenOrb development is guided by the needs of research projects and publications.
- MG does *NOT* have time or personnel to work on adding features to the sw unless it is “reimbursed” in a way acceptable to funding agencies.
  - Collaboration on a research project that leads to a co-authored publication is okay and work related to LSST should fulfill this criteria naturally.
- Also recall that OpenOrb is open-source sw which implies that *anyone* is free to improve and add features to the sw, and MG is happy to accept proposed improvements to the sw.
- Having said the above, we still need to identify the needs of the LSST community, so let MG know if there are needs that you think OpenOrb could fulfill.

Some identified needs:

- Documentation.
- Not currently modeling non-gravs, but MG looking for student to start working on it.
- Python bindings available for a very limited set of functions/capabilities.
- Transfer from Fortran 90/95 to Fortran 2003 or newer (natively object-oriented language).

Demonstration – Python

- example cases are found at
  - https://github.com/rhiannonlynne/notebooks/blob/master/PyOorb%20Demo.ipynb
  - https://github.com/oorb/oorb/tree/master/python

Demonstration – command-line

- example cases are found at
  - https://github.com/oorb/oorb/wiki
  - https://github.com/oorb/oorb/blob/master/doc/OpenOrb_Tutorial.tex

Information & help

- https://groups.google.com/g/oorb
- https://github.com/oorb/oorb/issues