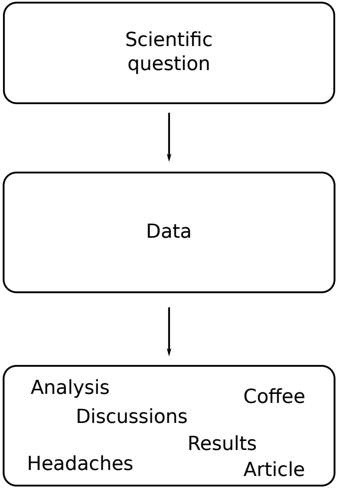
Online tools for planetary sciences

B. Carry[1] & M. Mahlke[2]
[1]Lagrange, Observatoire de la Côte d'Azur
[1]Institut d'Astrophysique Spatiale

A typical research project



A typical research project

Scientific question

↓

Data

↓

Analysis
Discussions
Coffee
Results
Headaches
Article

Repetitive (and tedious) tasks!

- **Planning and conduction of observations**
  - Observations already exist?
  - Target/sample available? visible?
- **Gathering ancillary data for the analysis**
  - Complementary information diameter, fall/find, ...
  - Context for research another population
- **Repetitive low-level analysis**
  - Spectral classification
  - Cross-matches & merges

Shared resources save community time

- **Tedious tasks? Share the load!**
  - Many agencies have the mission to support the community
    ESO/ESA/NASA, JPL/MPC/IMCCE, ...
  - The expertize is in the community → individual initiatives
    SSHADE, Meteoretical Bulletin, SMASS
  - ▶ More time for your research

- **Tedious tasks? Automatize them!**
  - Click, click, click... copy-paste, click...
  - Or code some processes to work for you
  - ▶ Virtual Observatory & Community librairies

- **Community services are less prone to errors!**
  - One user → one α-, β-tester, user...
  - Many users → bug reports! and community solutions & patches!
  - ▶ Robustness of analysis → results

Pointing a telescope

Example

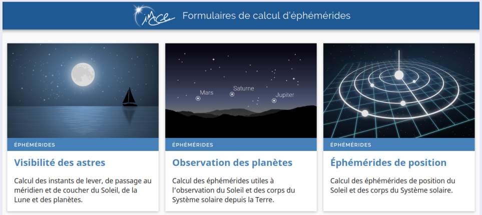
Where do I point the telescope from the name of a target?

Pointing a telescope

Example

Where do I point the telescope from the name of a target?

Answer: CDS, IMCCE Miriade, JPL SSD, MPC, Lowell AstEph



Visibility of targets

Example

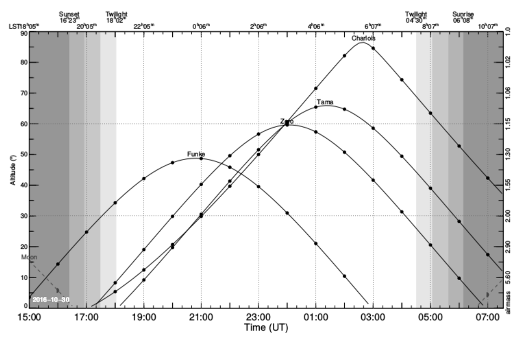
Can I observe asteroids Ceres, Pallas, 4321 tonight? And M31?

Visibility of targets

Example

Can I observe asteroids Ceres, Pallas, 4321 tonight? And M31?

Answer: IMCCE ViSiON, Lowell AstObs



Accessing data

Example

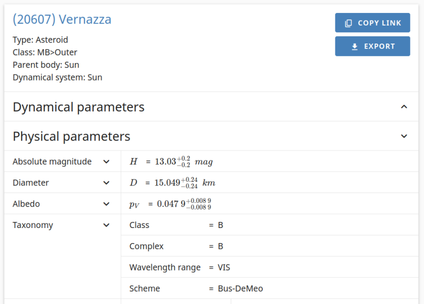
What is the taxonomy of Vernazza? the diameter of Groussin?

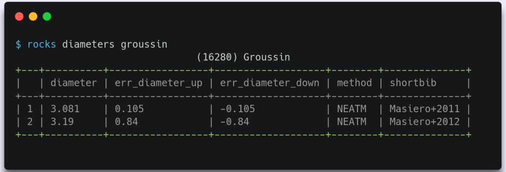
Accessing data

Example

What is the taxonomy of Vernazza? the diameter of Groussin?

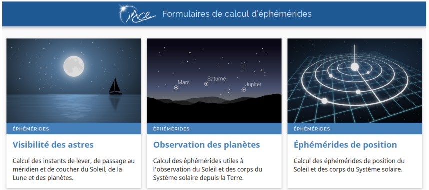
Answer: IMCCE SsODNet, JPL sbdb, OCA MP3C, Lowell AstInfo, SiMDA





Pimp my processing

- **Web forms** Access at human-scale
  - Reprocess archival observations
  - Need to contextualize and complement
  - Perform operations beyond our confort zone
- **Shared libraries** Automatize and rationalize
  - Local installation & calls
  - Part of codes, scripts → repeatability
- **Web services and APIs** Use remote resources
  - Send query & get answer
  - Maintenance on the provider side

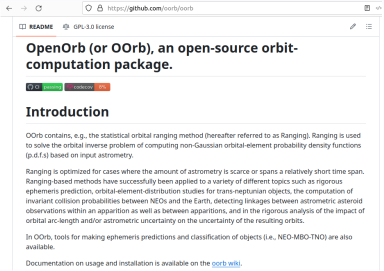


Pimp my processing

- **Web forms** Access at human-scale
  - Reprocess archival observations
  - Need to contextualize and complement
  - Perform operations beyond our confort zone
- **Shared libraries** Automatize and rationalize
  - Local installation & calls
  - Part of codes, scripts → repeatability
- **Web services and APIs** Use remote resources
  - Send query & get answer
  - Maintenance on the provider side

Pimp my processing

- **Web forms** Access at human-scale
  - Reprocess archival observations
  - Need to contextualize and complement
  - Perform operations beyond our confort zone
- **Shared libraries** Automatize and rationalize
  - Local installation & calls
  - Part of codes, scripts → repeatability
- **Web services and APIs** Use remote resources
  - Send query & get answer
  - Maintenance on the provider side



https://github.com/oorb/oorb

README GPL-3.0 license

OpenOrb (or OOrb), an open-source orbit-computation package.

CI passing codecov 8%

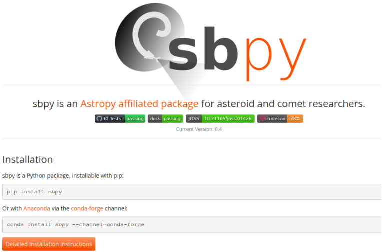
Introduction

OOrb contains, e.g., the statistical orbital ranging method (hereafter referred to as Ranging). Ranging is used to solve the orbital inverse problem of computing non-Gaussian orbital-element probability density functions (p.d.f.s) based on input astrometry.

Ranging is optimized for cases where the amount of astrometry is scarce or spans a relatively short time span. Ranging-based methods have successfully been applied to a variety of different topics such as rigorous ephemeris prediction, orbital-element-distribution studies for trans-neptunian objects, the computation of invariant collision probabilities between NEOs and the Earth, detecting linkages between astrometric asteroid observations within an apparition as well as between apparitions, and in the rigorous analysis of the impact of orbital arc-length and/or astrometric uncertainty on the uncertainty of the resulting orbits.

In OOrb, tools for making ephemeris predictions and classification of objects (i.e., NEO-MBO-TNO) are also available.

Documentation on usage and installation is available on the oorb wiki.



sbpy

sbpy is an Astropy affiliated package for asteroid and comet researchers.

CI Tests passing docs passing JOSS 10.21105/joss.01426 codecov 78%

Current Version: 0.4

Installation

sbpy is a Python package, installable with pip:

```
pip install sbpy
```

Or with Anaconda via the conda-forge channel:

```
conda install sbpy --channel=conda-forge
```

Detailed Installation Instructions

Pimp my processing

- **Web forms** Access at human-scale
  - Reprocess archival observations
  - Need to contextualize and complement
  - Perform operations beyond our confort zone
- **Shared libraries** Automatize and rationalize
  - Local installation & calls
  - Part of codes, scripts → repeatability
- **Web services and APIs** Use remote resources
  - Send query & get answer
  - Maintenance on the provider side

Reddened solar spectrum

sbpy's `sbpy.spectroscopy.SpectralGradient` can be used to redden a solar spectrum, although this approach may be revised with future sbpy capabilities.

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import astropy.units as u
from sbpy.spectroscopy import SpectralGradient
from sbpy.units import hundred_nm
from sbpy.calib import Sun
```

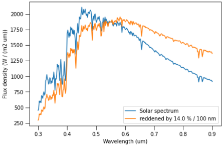
In [2]:
```
sun = Sun.from_builtin('E490_2014LR')
wave = np.linspace(0.3, 0.9, 300) * u.um
fluxd = sun.observe(wave)
S = SpectralGradient(14 * u.percent / hundred_nm, wave0=0.55 * u.um)
print(S)
```

```
WARNING: Source spectrum is tapered. [synphot.observation]
WARNING:astropy:Source spectrum is tapered.
14.0 % / 100 nm
```

`SpectralGradient` works with two-band photometry expressed as magnitudes. To use it for a spectrum, the function `to_color()` must be repeatedly called:

In [3]:
```
color_index = u.Quantity([S.to_color((0.55 * u.um, w)) for w in wave])
# express in linear units (reflectance)
r = 10**(0.4 * color_index.value)
```

In [4]:
```
ax = plt.gca()
ax.plot(wave, fluxd, label='Solar spectrum')
ax.plot(wave, r * fluxd, label='reddened by {}'.format(S))
plt.setp(ax, ylabel='Flux density ({})'.format(fluxd.unit),
        xlabel='Wavelength ({})'.format(wave.unit))
plt.legend()
plt.show()
```



Pimp my processing

- **Web forms** Access at human-scale
  - Reprocess archival observations
  - Need to contextualize and complement
  - Perform operations beyond our confort zone

```
>>> from astroquery.mpc import MPC
>>> result = MPC.query_object('asteroid', number=56788)
>>> result[0]['name'], result[0]['tisserand_jupiter']
    ('Guilbertlepoutre', 3.2)
```

- **Shared libraries** Automatize and rationalize
  - Local installation & calls
  - Part of codes, scripts → repeatability

- **Web services and APIs** Use remote resources
  - Send `query` & `get answer`
  - Maintenance on the provider side

Typical tasks and some solutions

TBD