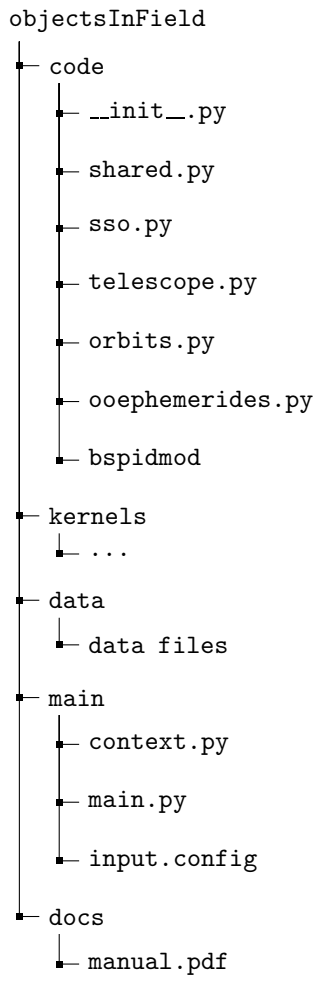
1 Introduction

This software generates a list of candidate detections for an input asteroid population and a list of telescope field pointings.

2 Installation

The software has been tested with Python 3.4 and 3.7. It requires spiceypy [1, 2], pyoorb [3], and other standard python packages.

The directory structure is as follows:



An additional directory will be created to store trajectories of asteroids as SPK files [1].

3 Inputs

The software should be run from the main directory. The general usage of the software is:

```
$ main.py configfile > output
or
$ main.py -f configfile > output
```

Here `configfile` is a file containing all the input parameters for the simulation. The output can be redirected to a file using the redirection operator, `>`. By default, the software doesn't overwrite existing asteroid SPK files unless you use the `-f` flag. A sample configfile named `input.config` is provided with the package.

The descriptions of the various files needed by the software are given in the subsections that follow.

3.1 Configfile

The 'configfile' follows the structure used by the *configparser* module of *Python*. A sample configfile named `input.config` is provided in the `main` directory. Briefly, the file is divided into sections that begin with headers of the form `[SECTION]`, where `SECTION` is the name of the section. Each section consists of multiple lines of the form `keyword = value`. The `configfile` for this software consists of four sections. All sections are mandatory but some keyword/value pairs are optional. Table 1 describes each keyword that the software accepts. Mandatory fields are denoted by an asterisk (*).

Keyword	Section	Default value	Description
`Data path` *	CONF	NA	Path of the directory where the SPICE meta-kernel, the JPL planetary ephemeris file, and the `Population model` file reside. The software also uses this folder to write the temporary files.
`SPICE metakernel` *	CONF	NA	Filename of the NAIF SPICE meta-kernel. The meta kernel resides in `Data path` and contains paths to all the common SPICE Kernels required by the software that are not survey specific. Read about how to prepare the meta-kernels at [1].

`Planetary ephem` *	CONF	NA	Name of the JPL planetary ephemeris file required by OpenOrb [3].
`nProc`	CONF	1	(Currently not supported) Number of parallel processes to use.
`Camera` *	CAMERA	NA	Name of the file containing the definition of the camera field of view. File description in Section 3.3.
`SPICE IK`	CAMERA	camera.ti	Filename of the SPICE instrument kernel that the software will write in the `Data path` folder.
`Threshold`	CAMERA	5	Angular separation threshold in degrees for performing an initial rough search for detectable targets. This threshold should be larger than the field of view of the camera.
`Population model` *	ASTEROID	NA	Name of the file containing the asteroid orbits. This file should be in the `Data path` folder.
`Asteroid SPK path` *	ASTEROID	NA	Path of the directory where all the asteroid SPKs are or will be stored. This path is relative to `Data path`.
`Asteroid SPKs` *	ASTEROID	NA	Asteroid SPK base filename. The trajectory of each asteroid will be saved in a separate SPK file having a filename that is the base name appended by an integer and the file extension (.bsp).
`Object1`	ASTEROID	1	Line number of the first asteroid orbit to be read from the `Population model` file. Should be > 1.
`nObjects`	ASTEROID	1	Number of asteroid orbits to read from the `Population model` file starting from `Object1`.

`Make SPKs`	ASTEROID	T	Flag to indicate whether the asteroid orbits should be propagated and the ephemerides should be saved in SPK files. If the appropriate SPK files already exists, set the value to F otherwise set it to T.
`SPK T0` *	ASTEROID	NA	Start time for generating the asteroids and/or the observatory spks.
`nDays`	ASTEROID	4000	Required length in days of asteroid and/or observatory spks.
`SPK step`	ASTEROID	20	Step size to use for generating the asteroid ephemerides.
`nbody`	ASTEROID	F	Type of propagation to generate asteroid SPKs (T for N-body and F for 2-body). 2-body just includes the Sun. N-body includes the 8 planets and the Moon.
`Survey database` *	SURVEY	NA	Name and path of the sqlite database that contains the field pointing information. The path should be given relative to `Data path`.
`Field1`	SURVEY	1	First field to read from the database
`nFields`	SURVEY	1000	Number of fields to read from the database. If `nFields` is greater than the number of fields in the database, then all fields will be read.
`Space`	SURVEY	F	Value of T indicates that the user is simulating a space-based survey. If `Space`=T, `SCID` is required and the spacecraft SPK file (named `SCID.bsp`) must be provided. If `Space`=F, `MPCobscode file` and `Telescope` are required.

`SCID` [a]	SURVEY	NA	Integer ID of the spacecraft. Required if `Space` = T. (See rules on SCID in section xx).
`MPCobscode file` [b]	SURVEY	NA	File containing the MPC observatory codes. Required if `Space`=F. File should reside in `Data path`
`Telescope` [b]	SURVEY	NA	MPC observatory code of the survey telescope. Required if `Space`=F.

Table 1: Configfile parameters. * indicates that the field is mandatory. [a]indicates mandatory if Space = T. [b] indicates mandatory if Space = F.

3.2 Survey database

The survey database is an LSST OpSim SQLite database or a reduced version of it with only the necessary fields. At a minimum the database requires the 'Summary' table with the fields described in Table 2.

Column name	Type	Units	Description
observationId	integer	-	Unique visit identifier.
ra	real	radians	Right Ascension (J2000) of the field center for this visit.
dec	real	radians	Declination (J2000) of the field center for this visit (same as Field.fieldDec).
observationstartMJD	real	days	Modified Julian Date at the start of a visit.
angle	real	radians	The orientation of the sky in the focal plane measured as the angle between North on the sky and the "up" direction in the focal plane.

Table 2: Survey database 'Summary' table.

3.3 Instrument field of view definition

The instrument field of view (FOV) is defined in a plain text file called the 'camera file', which is input via the `Camera` keyword in the `configfile`. The first line of the camera file defines the type of FOV. Currently allowed values are `Circle` or `Polygon`. If the first line is `Circle`, then the following line is the radius of the circle in degrees. If the first line is `Polygon`, then the subsequent lines will be the angular coordinates of the corners of the polygon in degrees. Figure 1 shows the coordinates for a rectangular FOV. Each row in the file lists the coordinates of a single corner. The corners are listed in either clockwise or counter-clockwise order. For the instrument shown in Figure 1, the camera file would be:

```
Polygon
x1 y1
x2 y2
x3 y3
x4 y4
```

The camera file for an instrument with a circular field of view with half angle of 1.5 degrees would be:

```
Circle
1.5
```

3.4 Asteroid population model

The Asteroid population model file is a plain text file containing a table of orbital elements in the Ecliptic J2000 frame. The first line of the file is the header describing the columns, which can be in any order. The suggested column names are as follows:

'OID' for Object ID,
'FORMAT' for format of elements. Suggested format is 'COM' for cometary elements,
'q' for perihelion distance in au,
'e' for eccentricity,
'i' for inclination in degrees,
'node' for longitude of ascending node in degrees,
'argperi' for argument of pericenter in degrees,
't_p' for time of pericenter passage in MJD,
't_0' for epoch of the orbital elements in MJD,

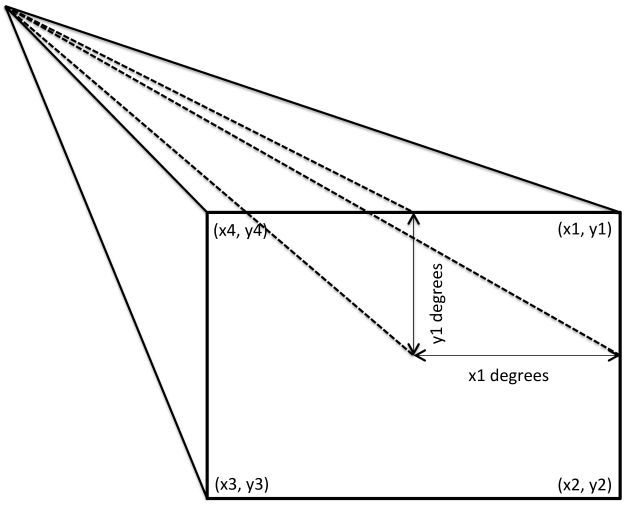


Figure 1: Coordinates of a rectangular field of view.

'H' for absolute magnitude.

Value of OID should be unique for each object.

References

[1] "Spice kernel required reading." [Online]. Available: ftp://naif.jpl.nasa.gov/pub/naif/toolkit_docs/C/req/kernel.html

[2] A. Annex, B. Carcich, J. McAuliffe, S. ya Murakami, S. Kulumani, M. de Val-Borro, and J. D. del Rio, "Andrewannex/spiceypy: Spiceypy 2.1.0," Nov. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1044745

[3] M. Granvik, J. Virtanen, D. Oszkiewicz, and K. Muinonen, "OpenOrb: Open-source asteroid orbit computation software including statistical ranging," *Meteoritics and Planetary Science*, vol. 44, pp. 1853–1861, Jan. 2009.