Serendipitous asteroid observation

Internship

USER MANUAL

SRON

February 23, 2018

by

Lenka Husárová

Contents

1. Introduction

In this manual, the code made during the internship in SRON, is described and explained. The code's functionality is to determine which asteroids are in the field of view of the specified instrument at the specified time. The asteroids found and their physical characteristics, as known from the JPL HORIZONS System [1], are saved in MySQL database. Using these information, the expected brightness (thermal emission) of each asteroid is calculated and saved in the database as well.

This tool uses an input table, Chapter 3, with data about the desired date, instrument and field of view to run. The tool has 4 functions. First one, goes over all the rows in the input table and runs the code for the ones which have not been updated yet. For the second function, a date is specified. If the data in the MySQL table were updated before this date, the code is run for them again. Third one, updates the whole input table, creates new tables and updates all the old ones. Last, is similar to third, but a date is specified and the code creates new tables but updates only the outdated ones.

The principle of the tool is fairly simple. The inputs from the input table are taken and fed in the Asteroid & Comet Field-of-View Search [2] made by JPL Solar System Dynamics Group. This allows the user to identify known comets or asteroids that are present (and may be visible) in the instrument's field of view. This system returns a table with the objects found and some of their characteristics, such as their position (RA,DEC) or the uncertainty in the positional prediction. The dynamics are from a high-precision n-body numerical integration with perturbations of other objects. The catalogued asteroids data are taken from the JPL HORIZONS System [1] and as new asteroids are detected daily and old data are updated, it is useful to rerun the code every few weeks. These information from the Asteroid & Comet Field-of-View Search System are then imported by the code into MySQL tables. Each table has a specific name, so called observation id, which specifies the observation date, instrument and field of view. More data from the JPL HORIZONS System are imported into the table as well and then these are used to calculate the expected brightness (thermal emission) of the asteroid at different wavelengths. This is of course saved in the MySQL table as well.

After the database with tables for each observation id is created, it is possible with the help of MySQL [3] to examine the data. The user can compare different tables together, or inspect in how many tables a specific asteroid appears. In general, this tool can be useful when interpreting data from an instrument. The database can be used to see which asteroids could be in the field of view and what brightness they should have at the specified wavelength according to what is known about them. Knowing this, and having the actual data from the instrument, the information for each of the asteroids can be updated to present better physical characteristics.

This manual consists of 4 chapters. In Chapter 2, the tool requirements are stated. The inputs for the code are discussed in Chapter 3 and Chapter 4 gives a short overview of how to run the code and what are the expected outputs.

2. Tool requirements

This chapter elaborates on the required programs for the tool to work. It also specifies all the needed packages and functions.

2.1 Programs needed

As this code is made in Python 3.2.6., it is recommended to have installed Python version 3.2.6. or higher. As in the older version some functions might not work as desired. Also in version 2, the syntax is different than in version 3 and this would lead to errors in the code. The installation is done preferably with the help of Anaconda [4].

All the information about the asteroid are saved in MySQL database, and therefore MySQL [3] needs to be installed as well. It is useful to have MySQL Workbench as this has a nice user interface and it is easy to examine the tables in the database. However, this is not necessary, as all the needed can be done trough the command prompt as well.

The Neatm function, which calculates the thermal emission of asteroids, is developed in C++, therefore a C++ compiler is required to compile the neatm function.

2.2 Packages

For Python, multiple packages are needed to run the code as desired. These packages are listed below and can be easily installed with the help of anaconda prompt, either using command *conda install package name* or *pip install package name*. Some of the packages could be already installed in the Anaconda environment by default.

- astropy (package focused on astronomy and astrophysics)
- callhorizons (package with Python interface to access JPL HORIZONS ephemerides and orbital elements)
- DateTime (package for helpful date format, able to work with time zones)
- matplotlib - optional (package to create plots and figures)
- numpy (package for scientific computing)
- pandas (package for doing practical, real world data analysis in Python)
- pymysql (package which provides the connection between Python and MySQL)
- scipy (package for scientific computing and technical computing)

2.3 Other functions

Neatm function is a function developed by Migo Mueller [5]. It calculates the thermal emission of an asteroid following the Near-Earth Asteroid Thermal Model [6].

To be able to use the code, few steps needs to be taken to install the Neatm function.
First, download the source code from [5]. Second, run 'make' in this directory, this requires C++ compiler g++. Then, move the resulting binary (Neatm) to a directory within your $PATH. Now the Neatm function should be accessible by the python code. If, for some reason, Python still cannot find Neatm during the run of the code, it is possible to make it work by moving the Neatm executable to the same folder as the script.

3. Inputs

In this chapter, the input requirements, such as their format, are discussed. Also the input data table is described. Before running the code, the next chapter, Chapter 4, should be read.

3.1 Input data table

To be able to use this tool, an input data table needs to be created first. This can be done manually in MySQL, but is recommended to do with the help of script called Create_Dummy_Database.py with the input file (Login_Data.txt). It creates a table with 15 columns and 3 rows of dummy data with a table name INPUT_Table. The dummy data can be changed inside the python scrip, or just deleted after the table is made by calling the command in the command prompt:
TRUNCATE TABLE table name;

New input data can be added into the input table by running the Import_New_Data_To_Input_Table _MySQL.py script with Login_Data.txt and Add_Data.txt as inputs. This can be done in Python by calling:
Import_New_Data_To_Input_Table_MySQL.Import_new_data_into_input_table('Login_Data.txt','Add_data.txt')
after importing Import_New_Data_To_Input_Table_MySQL or from the command line by calling the file name. This adds data into a table called INPUT_Table. Therefore, if the user changed the name of the input table it will create an error.

The format of the table is 15 columns, as seen in Figure 3.1: OBS_id, Observation_date, Instrument, Mode, Vertex1_1, Vertex1_2, Vertex2_1, Vertex2_2, Vertex3_1, Vertex3_2, Vertex4_1, Vertex4_2, Status, Number_of_Asteroids _Detected, Time_Updated_UTC
The OBS_id is a specific id made up of the field of view, date and instrument, but can be also chosen at random. This is the primary key of the table.

OBS_id	Observation_date	Instrument	Mode	Vertex1_1	Vertex1_2	Vertex2_1	Vertex2_2	Vertex3_1	Vertex3_2	Vertex4_1	Vertex4_2	Status	Num...	Time_Updated_UTC
170309_134500_M	2017/03/09 13:45:00 UTC	MIRI	NULL	22 deg	24 deg	2 deg	NULL	NULL	NULL	NULL	NULL	UPDATED	133	2018-02-15 14:53:34 UTC
170809_134500_NS	2017/08/09 13:45:00 UTC	NIRSpec	NULL	22 deg	24 deg	22 deg	24.2 deg	22.1 deg	24.3 deg	22.4 deg	24 deg	UPDATED	0	2018-02-15 14:54:26 UTC
181209_134500_M	2018/12/09 13:45:00 UTC	MIRI	NULL	22 deg	24 deg	2 deg	NULL	0.2 deg	NULL	8 deg	NULL	UPDATED	2	2018-02-15 14:55:44 UTC
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 3.1: The dummy Input table

Observation_date is the date and time as which the sky is observed. This is given in the format as shown in Figure 3.1 with the time zone specified. Instrument and mode are information about which instrument at which mode was used.

The Vertex 1-4 are the four vertices of the field of view. Vertex1_1 and Vertex1_2 specify the RA and DEC for Vertex1. It is possible to use 3 different shapes for the field of view, polygon, circle and rectangle. Polygon consists of 4 vertices, each of RA and DEC. Circle has a field of view center (RA,DEC) and an angle specifying the radius. Rectangle had a field of view center (RA,DEC) and 3 angles. The values should be given with units as well, as can be seen in Figure 3.1

Status is NULL at first, but after the tool is used, it is updated to UPDATED and the asteroids found in the field of view are counted and given in the next column.

The Time_Updated_UTC gives the date and time when the row was updated in UTC. This is used when the function to update outdated data is chosen.

3.2 Input text files

For this tool, 3 text input files are needed. First one is the Login_Data.txt. This file provides the user name and password to log in the MySQL database. The format is as can be seen in Login_Data.txt the text file:
User name: asteroid
Password: 34GH2B.

Second file consists of the input data for the tool to run. These are beaming parameter, albedo and relative reflectance. These values are needed for the brightness calculations and are not very well known for each asteroid, therefore at this point are specified for all asteroids as same. The format is as can be seen in Input_Data.txt the text file:
Beaming parameter: 1.3
Albedo: 0.2
Relative reflectance: 1.4

The third text file, Add_data.txt, adds more data in the input table, as discussed above.
The format is as can be seen in Add_Data.txt:
OBS_id Observation_date Instrument Mode Vertex1_1 Vertex1_2 Vertex2_1 Vertex2_2 Vertex3_1 Vertex3_2 Vertex4_1 Vertex4_2
200309_134500_NC, 2020/03/09 13:45:00 UTC, NIRCam, NULL, 43 arcmin, 56 arcsec, 0.3 arcsec, NULL, NULL, NULL, NULL, NULL, NULL
The first line are the names of the variables and the second are the input data. In this example, the field of view is circle, the field of view center RA and DEC and then the angle giving the radius. The empty values need to say NULL, to specify that they are empty.

4. Running the tool

In this chapter, few steps before running the tool are explained. Then, the process to run the tool is shortly shown and the expected output is described.

4.1 Steps before running the tool

First, to be able to run the tool properly, all the input files, as explained in Chapter 3, have to be altered as needed by the user and imported in the same folder as the Python scrips.

Second, the input data table has to be made before running the tool. This is can be done by running the function in the script called Create_Dummy_Database.py with the input file (Login_Data.txt). This will create a dummy database with 3 input rows. It is possible to run the code from the command line by calling the file name, or from Python by calling:
Create_Dummy_Database.Create_dummy_input_Database('Login_Data.txt')
after importing Create_Dummy_Database.

Third, it is recommended to make a new user for MySQL database instead of using the preprogrammed one. Creating a new user can be done trough the MySQL workbench or also the command prompt or MySQL shell, as explained in this article [7]. The user should be granted all the privileges and should be created at the localhost. Then the new user name and password should be updated in the input file (Login_Data.txt).

Fourth, at this point, the mode of the instrument is left blank in the input table. This is where the wavelength would be determined. This means that the wavelength (microns) needs to be specified, either as value or a list of values, to determine brightness at these values. This is done in the script called Run_Script_Tomake_Database.py by giving values to a variable lambdaMu.

4.2 How to run the tool

When all the previous steps has been taken, the tool can be run. This can be done from the script called Run_Script_Tomake_Database.py. As there are 4 functions, user can choose a one of them. This is done by uncommenting the desired line and updating the inputs for the function. This refers to the date, which determines which data needs to be rerun as outdated. The names of the input files or the lamdaMu input does not need to be changed unless the user renamed them.

4.3 Expected outputs

After the tool was run, the output should be a multiple tables in the MySQL database called ISPY. Each of these tables gives the asteroids expected in the specified field of view at the specified moment. For each asteroid, some physical characteristics and calculated brightness are given as well as can be seen in Figure 4.1 and 4.2. The table names are taken as the OBD_id from the INPUT_Table.

There are 26 columns, plus column or columns for brightness, this number depends on the number of values for wavelengths.

- OBD_id is a unique id made up of the field of view, date and instrument as specified in the input table as well. This is the foreign key for each table. Foreign keys help to cross-reference related data across tables.
- JPL_SPKID is a unique numeric id within JPL's SPICE system for the object.
- IAU_Number is the object's IAU number, if given.
- Name_Designation is the object's IAU name (if assigned) or primary designation.

OBS_id	JPL_SPKID	IAU_Number	Name_designation	RA	DEC1	Amag	dRAcosD	dDEC_by_dt	CntDst	PsAng	Data_Arc	Nobs
170309_134500_M	2034708	34708	Grasset	01:27:18.71	+25 39 02.2	18.0	89.15	20.63	5968.7	354.6	1986-2018	819
170309_134500_M	2056307	56307	(1999 RY125)	01:33:18.62	+23 46 32.6	19.9	44.22	13.81	4443.9	100.2	1997-2018	754
170309_134500_M	2072942	72942	(2002 CU9)	01:30:30.89	+22 39 23.0	20.8	81.09	35.90	5264.6	156.6	1995-2016	363
170309_134500_M	2076865	76865	(2000 XW38)	01:33:40.88	+23 17 58.5	19.7	76.03	20.47	5319.3	118.0	1994-2018	924
170309_134500_M	2086156	86156	(1999 RB203)	01:29:09.10	+22 55 39.5	18.9	59.50	17.05	3975.9	166.1	1999-2018	943
170309_134500_M	2086429	86429	(2000 BJ29)	01:30:37.33	+23 23 06.6	19.2	56.95	18.33	3093.4	135.6	1999-2018	742
170309_134500_M	2091405	91405	(1999 NB10)	01:30:36.61	+22 19 21.0	19.2	80.21	25.51	6413.5	160.2	1984-2018	696
170309_134500_M	2095461	95461	(2002 DZ2)	01:30:24.17	+25 02 33.5	20.4	43.80	17.14	4237.8	27.5	1997-2017	448
170309_134500_M	2105252	105252	(2000 QD7)	01:21:06.12	+23 04 50.5	20.9	75.19	31.08	6583.7	240.2	1960-2018	323
170309_134500_M	2106078	106078	(2000 SK347)	01:32:50.66	+23 36 33.4	19.8	61.19	19.80	4229.7	109.2	1994-2018	587

Figure 4.1: The expected Output table, part 1

Nobs	SMAA_3sig	SMIA_3sig	Theta	Pixel_x	Pixel_y	H	G	alpha	r	delta	eta	pv	Relative_reflectance	Brightness_3.6Micron
819	0.121311	0.0859201	9.5	NULL	NULL	13.6	0.15	21.6177	1.867246966133	2.46332445945708	1.3	0.2	1.4	0.12338774907998461 mJy
754	0.0761838	0.0711297	14.3	NULL	NULL	13.5	0.15	11.6568	3.400437624657	4.05862809441922	1.3	0.2	1.4	0.017449746535917927 mJy
363	0.0767111	0.0647549	34.5	NULL	NULL	16.5	0.15	21.6136	1.82397608626	2.43801809078357	1.3	0.2	1.4	0.009302051497555171 mJy
924	0.0687685	0.0648275	-15.3	NULL	NULL	15.0	0.15	19.515	2.04848894661	2.66165368281724	1.3	0.2	1.4	0.02385412337197746 mJy
943	0.0660895	0.0523271	19.1	NULL	NULL	13.3	0.15	14.3923	2.698252918767	3.3569532962597	1.3	0.2	1.4	0.04476071548207282 mJy
742	0.0696427	0.0657874	13.4	NULL	NULL	13.5	0.15	14.0794	2.785870233434	3.43915984729029	1.3	0.2	1.4	0.033543460012068 1 mJy
696	0.0897725	0.0734952	13.6	NULL	NULL	14.5	0.15	19.0711	2.04916378464	2.68106012581247	1.3	0.2	1.4	0.03775765492293044 mJy
448	0.0737215	0.0675731	39.5	NULL	NULL	14.1	0.15	12.167	3.27413268949	3.92604089470262	1.3	0.2	1.4	0.011328047599564024 mJy
323	0.0757471	0.07493	-79.4	NULL	NULL	16.4	0.15	19.4112	1.963801472262	2.61140090837936	1.3	0.2	1.4	0.0076601588514642 4 mJy
587	0.0591075	0.0550906	13.9	NULL	NULL	14.3	0.15	15.1647	2.617221793508	3.25660497337832	1.3	0.2	1.4	0.019791310395571304 mJy

Figure 4.2: The expected Output table, part 2

- RA is the predicted astrometric right-ascension (light-time corrected) of the object in ICRF/J2000 cooordinates. Units: HH MM SS.ff (hours, minutes, and seconds of time)
- DEC1 is the predicted astrometric declination (light-time aberrated) of the object in ICRF/J2000 coordinates. Units: DG MN SC.ff (degrees, minutes, and seconds of arc)
- Amag is the apparent visual magnitude of the object.
- dRAcosD is the instantaneous rate of change of astrometric right-ascension at the observation time. Units: ARCSECONDS PER HOUR
- dDEC_by_dt is the instantaneous rate of change of astrometric declination. Units: ARCSECONDS PER HOUR
- CntDst is the angular distance of the object from the centroid of the field of view. Units: ARCSECONDS
- PsAng is the position angle of the object relative to the field centroid, measured counter-clockwise (CCW) from Celestial North direction (e.g. line of constant right-ascension). Units: DEGREES
- Data_Arc are the years spanned by the astrometry included in the object's orbit solution. If less than one year, the number of days in the data arc is displayed.
- Nobs is the number of observations included in the object's orbit solution.
- SMAA_3sig is the angular width of the 3-sigma error ellipse semi-major axis in plane-of-sky. Units: ARCSECONDS
- SMIA_3sig is the angular width of the 3-sigma error ellipse semi-minor axis in plane-of-sky. Units: ARCSECONDS.
- Theta is the orientation angle of the error ellipse in plane-of-sky; the clockwise angle from the direction of increasing RA to the semi-major axis of the error ellipse, in the direction of increasing DEC. Units: DEGREES.
- Pixel_x are the pixels in x direction at which the asteroid should appear.
- Pixel_y are the pixels in y direction at which the asteroid should appear.
- H is the absolute magnitude of the asteroid.

- G is the slope parameter.
- alpha is the Sun-Target-Observer angle; the interior vertex angle at target center formed by a vector to the apparent center of the Sun at reflection time on the target and the apparent vector to the observer at print-time. Slightly different from true PHASE ANGLE (requestable separately) at the few arcsecond level in that it includes stellar aberration on the down-leg from target to observer. Units: DEGREES
- r is Heliocentric range of the target center at the instant light seen by the observer at print-time would have left the target center (print-time minus down-leg light-time). The Sun-to-target distance traveled by a ray of light emanating from the center of the Sun that reaches the target center point at some instant and is recordable by the observer one down-leg light-time later at print-time. Units: AU
- delta is range of target center with respect to the observer at the instant light seen by the observer at print-time would have left the target center (print-time minus down-leg light-time); the distance traveled by a light ray emanating from the center of the target and recorded by the observer at print-time. Units: AU
- eta is the beaming parameter of the asteroid.
- pv is the albedo of the asteroid.
- Relative_reflectance is the relative reflectance of the asteroid.
- Brightness_xMicron is the calculated brightness of the asteroid, both due to thermal emission and reflected brightness.

At this time, there is no calculation for Pixel x and y included in the code.

Each table has a comment which gives user the information about input data for this table, observation date, instrument, mode and the field of view. This comment can be found when running a command in the command prompt or MySQL shell:
SELECT TABLE_COMMENT FROM information_schema.TABLES WHERE TABLE_NAME = 'table name';
To see the table the following command can be used:
SELECT * from table name;
To examine the created tables in more ways, the user is recommend to read more about MySQL databases [3].

Bibliography

[1] NASA Jet Propulsion Laboratory, "Solar system dynamics," 2018, Cited 14-02-2018 from: https://ssd.jpl.nasa.gov/horizons.cgi.

[2] JPL Solar System Dynamics Group, "Asteroid & comet field-of-view search reques," 2018, Cited 19-02-2018 from: https://ssd.jpl.nasa.gov/x/ispy.html.

[3] MySQL, "Mysql," 2018, Cited 14-02-2018 from: https://www.mysql.com/.

[4] Anaconda, "Anaconda," 2018, Cited 14-02-2018 from: https://www.anaconda.com/.

[5] M. Mueller, "Neatm," 2018, Cited 15-02-2018 from: https://github.com/MigoMueller/NEATM.

[6] A. W.Harris, "A thermal model for near-earth asteroids," *Icarus*, vol. 131, no. 2, pp. 291–301, 1998.

[7] E. Sverdlov, "How to create a new user and grant permissions in mysql," 2012, Cited 15-02-2018 from: https://www.digitalocean.com/community/tutorials/how-to-create-a-new-user-and-grant-permissions-in-mysql.