User Manual

get_data(planet, observer,observatory,city, start, end, interval_mode, step_size, angular)
Get the data based on user-specified parameters.

Planet:
The name of the target you want to observe. e.g. "Earth","Venus", case insensitive.
observer:
The observer, we provide different options:

Sun:
Choose the observer to be at the center of the sun.
Earth:
Choose the observer to be at the center of the Earth.
observatory:
choose to generate data from an observatory.
city:
choose to generate data from a city.

observatory:
if observer="observatory", choose the location of the observatory. Available input can be found here: "https://ssd.jpl.nasa.gov/horizons.cgi\#top". The format should be [code,observatory name]
city:
if observer="city", choose the location of the city. Available input can be found here:"https://ssd.jpl.nasa.gov/horizons.cgi\#top".
start:
starting time of the ephemeris data. format should be {BC/AD} YYYY-MMM-DD {hh:mm},{} are optional
end:
end time of the ephemeris data.
interval_mode:
choose the unit of the time interval, available options are:

d: day
h: hour
m: minute
Y: year
MO: month

step_size:
step size of the time interval, with unit specified by interval_mode.
angular:
output data in Azimuth and elevation coordinates.

addnoise(data,mu,sigma,typ)
Add noise to data

data:
The ideal data in numpy array you want to add noise on.

mu:
The mean of the gaussian noise distribution or the center of the uniform noise.
sigma:
Standard deviation of the gaussian noise or the range of the uniform noise.
typ:
type of noise, options:"gaussian" or "uniform".

sparsify(data,percent)
Randomly draw some data out to mimic the effect of bad weather or just some missing data.
data:
Original data to be sparsified.
percent:
Percentage of the data to be kept.

visibledata(planettarget, planetob,start, end, interval_mode, step_size,Rsun=1/215)
Get rid of a data point when planettarget is ahead or behind the sun as observed on planetob so that it shouldn't be visible.
planettarget:
The target of the planet you want to observe.
planetob:
The observer planet
start,end,interval_mode,stepsize:
see description in get_data() function
Rsun:
Radius of the sun, in a.u. Unit.

c2s(cor)
Cartesian to spherical coordinates conversion.
Cor:
the cartesian coordinates
s2c(cor)
spherical to cartesian coordinates conversion
Cor:
the spherical coordinates.
outtxt(cor,outfile)
Output data into txt file
Cor:
Output coordinates data
Outfile:
Output file name

is_sun_2_vis(is_sun,thresh):
Return flags that determine whether planet could seen considering sun as the background. The visibility is based on “thresh”.

Is_sun:
numpy array of strings
Holds arrays containing a string with a single space ' ', 'A', 'N', 'C', or '*'. Meanings below
thresh: string
Sets the is_sun value(s) that constitute the object being visible.
This is mean to reflect the fact that most celestial bodies are not visible to the naked eye during the day
Designations below:

' ' (string of one space): Night time. Probably not useful, as this will set everything to not visible

'A': astronautical sunrise/dusk sun is well below horizon but is lighting up the sky somewhat
Setting this as threshold means any setting greater than or equal this (all that follow) will render the object invisible

'N': nautical sunrise/dusk = sun is still below the horizon, but sky is lighting up

'C': civil sunrise = sun is breaching over the horizon

'*': daytime using this setting means planets are visisble until daytime