

Celestial object visibility plots

Elyar Sedaghati

Level: Beginner → Intermediate

Package requirement: `numpy`, `matplotlib`, `astropy`

January 7, 2018

In this exercise we will write a simple code for coordinate transformations and the creation of visibility curves. It is planned to demonstrate the capabilities of the [astropy.coordinates](#) package. For more convenient and/or complex observation planning, you should consider the [astroplan](#) package.

Here are some hints that should help with designing the code. Please click on the coloured links for the documentation of that specific class/package.

- Import the basic necessary packages (`numpy`, `matplotlib`)
- Import packages for finding and transforming coordinates (look into the [astropy](#) package)
- The variables should ideally be soft-coded, where the user is asked to input the target name, the date of the observations and the observatory name.
- Provide the user with a list of available observatories to choose from in [astropy.coordinates.EarthLocation](#)
- Get the coordinates of the the target provided by the user ([astropy.coordinates.SkyCoord](#))
- Get the coordinates of the the telescope provided by the user ([astropy.coordinates.EarthLocation](#))
- Calculate alt, az coordinates of the target for the given observatory ([astropy.coordinates.AltAz](#) & the `transform_to` method in the `SkyCoord` class)
- Convert alt, az to airmass and plot it as a function of time (`secz` method in the `AltAz` class)
- Calculate alt, az values for both the sun and the moon during the night of the observations ([astropy.coordinates.get_sun](#) and [astropy.coordinates.get_moon](#))
- Remember to recalculate alt, az coordinates of the target for these new time stamps.
- Make the final visibility plot with `matplotlib`, which includes the altitude of the target, the moon and the sun, indicates twilight as gray-shaded and dark time as black-shaded. Optionally you can include the azimuth variations of the target as a `colorbar`. For a nicer-looking plot, I recommend styling the `matplotlib` graphs with the `astropy_mpl_style` method from the [astropy.visualization](#) package.