Crossmatching in Astropy

Here's an example of using Astropy to crossmatch two catalogues with 2 objects each:

```
from astropy.coordinates import SkyCoord
from astropy import units as u
coords1 = [[270, -30], [185, 15]]
coords2 = [[185, 20], [280, -30]]
sky_cat1 = SkyCoord(coords1*u.degree, frame='icrs')
sky_cat2 = SkyCoord(coords2*u.degree, frame='icrs')
closest_ids, closest_dists, closest_dists3d = sky_cat1.match_to_catalog_sky
print(closest_ids)
print(closest_dists)
```

The `SkyCoord` objects are general purpose sky catalogue storage and manipulation objects in Astropy. They take anything that looks like an array of coordinates as long as you specify the units (here we specify degrees with `u.degree`) and a reference frame (ICRS (https://en.wikipedia.org/wiki/International_Celestial_Reference_System) is essentially the same as equatorial coordinates. The outputs, `closest_id` and `closest_dists` give the matching object's row index in `sky_cat2` and the distance to it. `closest_dists` is the angular distance while `closest_dists3d` is the 3D distance which we're not concerned with here.

Note

Astropy returns distances as `Quantity` objects. You can convert these to NumPy arrays by accessing their `value` attribute like this:

```
closest_dists_array = closest_dists.value
```