# Lec14-astropy1-RyanSponzilli

October 10, 2024

# 1 ASTR 310 Lecture 14 - astropy1

### 1.0.1 Exercise 0: version check

Ideally you will have astropy 5.3 or higher. Astropy 5.1.0 does not play nicely with matplotlib 5.7.

If you don't have astropy at all, see www.astropy.org .

[1]:
```
import astropy
print('astropy', astropy.__version__)

import matplotlib
print('matplotlib', matplotlib.__version__)
```

```
astropy 6.1.0
matplotlib 3.8.4
```

### 1.0.2 Exercise 1: units and constants

Using Astropy, compute the following quantities. Hints on how to get started are in the reading for today!

**1.** The escape velocity (in km/s) of a $2.3M_\odot$ giant star with a radius of $150R_\odot$. [2 pts]

$$v_{esc} = \sqrt{\frac{2GM}{R}}$$

[20]:
```
from astropy import units as u
from astropy import constants as const
```

[21]:
```
v_esc = ((2*const.G*(2.3*u.solMass)/(150*u.solRad))**0.5).to(u.km / u.s)
print(v_esc)
```

```
76.48532239709607 km / s
```

**2.** The wavelength (in angstroms) of the peak of the spectrum of a 11,350 K blackbody. [2 pts]

$$\lambda_{peak} = \frac{0.29 \text{ cm K}}{T}$$

[33]:
```
lambda_peak = (0.29 * u.cm * u.K / (11350 * u.K)).to(u.Angstrom)
print(lambda_peak)
```

```
2555.0660792951535 Angstrom
```

**3.** The orbital period (in minutes) of a $1M_\odot$ white dwarf orbiting a $0.6M_\odot$ white dwarf at a separation of $10^5$ km. [2 pts]

$$P = 2\pi\sqrt{\frac{a^3}{G(M_1+M_2)}}$$

[32]:
```
period = (2*pi*((1e5 * u.km)**3 / (const.G * (1*u.solMass + 0.6*u.solMass)))**0.
↪5).to(u.minute)
print(period)
```

```
7.186423546252425 min
```

**4.** The Compton wavelength (in fm) of a nickel-56 nucleus, given its rest mass of 52.110 GeV $c^{-2}$. [2 pts]

$$\lambda = \frac{h}{mc}$$

[42]:
```
lambda_compton = (const.h / ((52.110 * u.GeV / const.c**2) * const.c)).to(u.fm)
print(lambda_compton)
```

```
0.02379278419366729 fm
```

**5.** The distance (in Mpc) to a radio galaxy with a flux density of 8400 Janskys (Jy) and a spectral luminosity of $6 \times 10^{35}$ erg $s^{-1}$ $Hz^{-1}$. What is a Jansky? 1 Jy = $10^{-26}$ W $m^{-2}$ $Hz^{-1}$. [2 pts]

$$F_\nu = \frac{L_\nu}{4\pi d^2}$$

[47]:
```
d = (((6e35 * u.erg / u.second / u.Hz) / (4*pi*8400*u.Jy))**0.5).to(u.Mpc)
print(d)
```

```
244.33208718480088 Mpc
```

### 1.0.3 Exercise 2: sky coordinate conversions and angles

Using `SkyCoord`, `EarthLocation`, and `Time` objects, perform the following calculations:

**1.** Find the angular distance between the Galactic center ($l = 0^\circ$, $b = 0^\circ$ in galactic coordinates) and the globular cluster M13. [3 pts]

[54]:
```
from astropy.coordinates import SkyCoord, EarthLocation
center = SkyCoord(0*u.degree,0*u.degree,frame="galactic")
cluster = SkyCoord.from_name("M13")
```

```python
print(center.separation(cluster))
```

```
67d06m01.5648835s
```

**2.** Find the angular distance between Venus and the Sun on February 29, 2024 at 23:00 UTC (i.e. "2024-02-29T23:00:00"). Use the `astropy.coordinates.get_body()` method to get sky coordinates for Solar System bodies at a given time. [3 pts]

[58]:
```python
import astropy.coordinates


venus = astropy.coordinates.get_body("Venus", astropy.time.Time("2024-02-29T23:
↪00:00"))
sun = astropy.coordinates.get_body("Sun", astropy.time.Time("2024-02-29T23:00:
↪00"))

print(venus.separation(sun))
```

```
24d24m56.515288s
```

**3.** Create an altitude-azimuth coordinate frame object (`AltAz`) corresponding to the location of the ALMA Observatory at the same time as above. Create a `SkyCoord` object corresponding to the sky position of the Large Magellanic Cloud (RA 05h23m34.5s, Dec $-69°45'22''$ in the `icrs` frame). Finally, using the `SkyCoord` object's `transform_to` method, find its representation in the alt-az frame you created. This will give the altitude and azimuth of the LMC at ALMA this evening at 8 pm local time. [4 pts]

[78]:
```python
from astropy.time import Time
from astropy.coordinates import AltAz

target = SkyCoord.from_name("LMC")
loc = EarthLocation(lat=-23.0235*u.degree, lon=-67.7539*u.degree,height=0*u.m)
time = Time("2024-02-29T23:00:00")

# Transform from Ra/Dec to Alt/Az
print(target.transform_to(AltAz(location=loc, obstime=time)))
```

```
<SkyCoord (AltAz: obstime=2024-02-29T23:00:00.000, location=(2223467.87192447,
-5435935.18940991, -2479114.72438978) m, pressure=0.0 hPa, temperature=0.0
deg_C, relative_humidity=0.0, obswl=1.0 micron): (az, alt) in deg
    (177.93289071, 43.21314132)>
```