Velocity & Galactic Rotation

In this notebook, you'll go through all the steps needed to go from a measured velocity to a measurement of the Galactic mass assuming circular orbits.

You just need to read along and execute cells for the first ~90%. Then, at the end, you'll repeat what you did in the top half, but for a different target.

Basic Python Setup

In [1]:
```
# preface - this is just stuff you need to run the code
%matplotlib inline
import pylab as pl
from astropy import units as u, constants, visualization
visualization.quantity_support()

# if you're using a white background, comment this next line out or delete it
# ("comment out" means put a # at the start of the line)
pl.style.use('dark_background')
```

In [2]:
```
# This cell loads the moduls you need to set up an observer, target, and time

from astroplan import Observer
from astropy import units as u, coordinates
from astropy.table import Table
import datetime
import pytz
tz = pytz.timezone('US/Eastern')
```

In [3]:
```
# This cell has been "commented out" using triple-quotes

# Right now, it doesn't do anything, but it's here to show you how you would load up your own observational
# data to make a target
"""
datatable = Table.read('data/psd_200725_100309_tint60s_sdr0_fsw.fits')
obslon = float(datatable.meta['--obs_lon']) * u.deg
obslat = float(datatable.meta['--obs_lat']) * u.deg
obs_time_notz = datetime.datetime.strptime(datatable.meta['DATE-OBS'], "%y%m%d_%H%M%S")
location = coordinates.EarthLocation.from_geodetic(obslon, obslat, 100*u.m)
"""
# if you wanted to use this code, you'd remove the """'s
```

Out[3]:
```
'\ndatatable = Table.read(\'data/psd_200725_100309_tint60s_sdr0_fsw.fits\')\nobslon = float(datatable.meta[\'--obs_lon\']) * u.
deg\nobslat = float(datatable.meta[\'--obs_lat\']) * u.deg\nobs_time_notz = datetime.datetime.strptime(datatable.meta[\'DATE-OB
S\'], "%y%m%d_%H%M%S")\nlocation = coordinates.EarthLocation.from_geodetic(obslon, obslat, 100*u.m)\n'
```

Observer / Observation Setup

In [4]:
```
# instead of using the "real data" from the previous cell, we're going to make up the data
# You can't do this in the lab!  This is just to show you how the code works
gainesville_location = coordinates.EarthLocation.from_geodetic(lon=-82.3*u.deg, lat=29.6*u.deg, height=100*u.m)
observer = Observer(gainesville_location, timezone='US/Eastern')

# we're using a made-up time
obs_time_notz = datetime.datetime(year=2020, month=10, day=6, hour=20, minute=50, second=13)
obs_time = tz.localize(obs_time_notz)

# we're using a made-up pointing direction
alt = 48*u.deg
az = 220*u.deg
target_altaz = coordinates.SkyCoord(alt=alt,
                                    az=az,
                                    location=observer.location,
                                    obstime=obs_time, frame='altaz')

# we want the coordinate in both altaz and galactic
target_galactic = target_altaz.galactic

target_altaz, target_galactic
```

Out[4]:
```
(<SkyCoord (AltAz: obstime=2020-10-07 00:50:13, location=(743674.13504561, -5500332.52750589, 3131946.80724394) m, pressure=0.0
hPa, temperature=0.0 deg_C, relative_humidity=0.0, obswl=1.0 micron): (az, alt) in deg
     (220., 48.)>,
 <SkyCoord (Galactic): (l, b) in deg
     (27.85023626, -0.02210464)>)
```

Radial Velocity Correction

We want to calculate the `radial_velocity_correction` , which transforms from "topocentric" (relative toour position on the surface of the Earth) to "barycentric" (relative to the center-of-mass of the solar system)

In [5]:
```
# our velocity in the direction of the target
target_galactic.radial_velocity_correction()
```

Out[5]: $-28357.686\ \frac{\mathrm{m}}{\mathrm{s}}$

This is how fast we (on the surface of Earth) are moving in this direction relative to the solar system's barycenter.

In [6]:
```
# What direction were we looking in?
# We can check this in both RA/Dec (ICRS) and Galactic
target_galactic.icrs, target_galactic.galactic
```

Out[6]:
```
(<SkyCoord (ICRS): (ra, dec) in deg
     (280.55720533, -4.5315173)>,
 <SkyCoord (Galactic): (l, b) in deg
     (27.85023626, -0.02210464)>)
```

Velocity Measurement -> Barycentric Conversion

In the lecture, at this stage, I re-fit a spectral line. We'll skip that now and use a made up line-of-sight velocity.

In [7]:
```
# this is our made-up line-of-sight velocity
# you will measure this from your data by fitting a Gaussian profile to the observed spectrum
# this observed velocity is in the *topocentric* frame
observed_line_of_sight_velocity = 30*u.km/u.s
```

To convert to *barycentric* coordinates, we use the `radial_velocity_correction` we calculated:

In [8]:
```
velocity_bary = observed_line_of_sight_velocity + target_galactic.radial_velocity_correction()
velocity_bary
```

Out[8]: $1.6423143\ \frac{\mathrm{km}}{\mathrm{s}}$

Convert to LSR coordinates

The sun isn't moving much relative to nearby stars.

We call the motion of our local region the "Local Standard of Rest" (LSR).

We need to measure the velocity in the LSR frame, otherwise we get a biased measurement of the Galactic rotation.

In [9]:
```
# we need to add velocity information to our SkyCoord
# because of an astropy bug, we have to add on some "dummy" values - the pm_ra_cosdec, pm_dec, and distance
# don't worry about these numbers, just use exactly what we put here
# otherwise, though, the target_altaz & velocity_bary variables are things you should define yourself
# Note that the velocity should be the barycentric velocity here!
skycoord_withvel = coordinates.SkyCoord(ra=target_altaz.icrs.ra, dec=target_altaz.icrs.dec,
                                        radial_velocity=velocity_bary,
                                        pm_ra_cosdec=0*u.arcmin/u.yr, pm_dec=0*u.arcmin/u.yr, distance=1*u.pc,
                                        frame='icrs')
skycoord_withvel
```

Out[9]:
```
<SkyCoord (ICRS): (ra, dec, distance) in (deg, deg, pc)
    (280.55720533, -4.5315173, 1.)
 (pm_ra_cosdec, pm_dec, radial_velocity) in (mas / yr, mas / yr, km / s)
    (0., 0., 1.64231431)>
```

Once we have a sky coordinate with velocity attached - and that velocity is barycentric - we can determine what the LSR velocity is by "transforming" to the LSR frame:

In [10]:
```
# we take the skycoord_withvel defined in the previous cell and "transform" it to LSR
# note that the skyframe was in the ICRS frame - this is important, since the frames
# all move with respect to one another
lsrcoord = skycoord_withvel.transform_to(coordinates.LSR)
lsrcoord.radial_velocity
```

Out[10]: $17.171886\ \frac{\mathrm{km}}{\mathrm{s}}$

Galactocentric Radius of the Tangent Point

We'll use "LSR" coordinates to calculate the rotation curve of the Galaxy.

In [11]:
```
# r0 is the distance between the sun and the Galactic center;
# it's somewhere between 8 and 8.5 kpc,
# but the IAU-recommended number is still 8.5 kpc
r0 = 8.5*u.kpc
```

Most of the gas you see when looking at the inner galaxy is "stacked up" at the tangent point, as illustrated below.

In [12]:
```
# don't run this cell unless you have the file mw_tangentpoint_geometry.png in your current directory
# (it will crash and then you won't have the diagram any more)
from IPython.display import Image
Image("mw_tangentpoint_geometry.png")
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\display.py in _data_and_metadata(self, always_both)
   1292         try:
-> 1293             b64_data = b2a_base64(self.data).decode('ascii')
   1294         except TypeError:

TypeError: a bytes-like object is required, not 'str'

During handling of the above exception, another exception occurred:

FileNotFoundError                         Traceback (most recent call last)
C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\formatters.py in __call__(self, obj, include, exclude)
    968 
    969             if method is not None:
--> 970                 return method(include=include, exclude=exclude)
    971             return None
    972         else:
```

The tangent point geometry tells you what Galactocentric distance you're observing.

For example, for $\ell = 30$, $r_0 = 8.5$ kpc, $r_{gal} = \sin(\ell) r_0 = 4.25$ kpc.

The velocity we observe is the circular velocity, v_{orb}, minus the sun's velocity projected in that direction.

However, the LSR is "at rest" with respect to the galaxy (to the best of our ability to measure it), so we've already subtracted that.

Orbital Velocity to Mass: Overview

Recall that the orbital velocity of a body in a gravitational potential only depends on the mass interior to that body.

$$v_{orb} = \sqrt{\frac{GM}{R}}$$

If we can measure v_{orb}, and we know what R we're observing, we can infer the contained mass, M.

$$M = \frac{R_{gal} v_{orb}^2}{G}$$

What mass do we infer?

In [13]:
```
# our longitude comes from the target we pointed at:
target_galactic.l
```

Out[13]: $27^{\circ}51'00.8505193''$

In [14]:
```
# it's in the midplane because b is about zero
# (if this number's not close to zero, you're not looking at the Galactic plane!)
target_galactic.b
```

Out[14]: $-0^{\circ}01'19.57670739''$

Galactic Longitude to Galactocentric Radius

What is R_{gal}? $R_{gal} = \sin \ell \times r_0$:

In [16]:
```
# calculate rgal, the distance from the galactic center to the tangent point along our target line-of-sight
import numpy as np
rgal = np.sin(target_galactic.l) * r0
rgal
```

Out[16]: 3.9708774 kpc

Our LSR velocity is the velocity we want to measure, so to get mass, we just put in:

$$M = \frac{R_{gal} v^2}{G}$$

In [17]:
```
# we'll put the observed radial velocity into its own variable, but we calculated it already above
vlsr = lsrcoord.radial_velocity
```

Orbital Velocity + Galactocentric Radius -> Mass

In [18]:
```
# calculate the mass
mass = rgal * vlsr**2 / constants.G
mass
```

Out[18]: $1.754352 \times 10^{13} \; \frac{\mathrm{km^2\,kg\,kpc}}{\mathrm{m^3}}$

That mass is in not-very-useful units. Convert it to Solar Masses:

In [19]:
```
mass.to(u.M_sun)
```

Out[19]: $2.7224591 \times 10^{8} \; \mathrm{M_{\odot}}$

So if we had done this observation, we would infer that the mass of the Galaxy within a radius of $r_{gal} = 4$ kpc is about 3×10^8 $\mathrm{M_{\odot}}$. Because the numbers above are made up, though, don't trust this measurement!

Exercise step

Repeat the above for an observation with the following properties:

altitude = 58°

azimuth = 220°

observation time: October 6, 2020 at 20:44:35

observed velocity: 27 km/s

In [23]:
```
# Step 1: Create your observer object
# (you can reuse the same observer)
# instead of using the "real data" from the previous cell, we're going to make up the data
# You can't do this in the lab!  This is just to show you how the code works
observer2_location = coordinates.EarthLocation.from_geodetic(lon=-82.3*u.deg, lat=29.6*u.deg, height=100*u.m)
observer2 = Observer(observer2_location, timezone='US/Eastern')
```

In [24]:
```
# Step 2: Create the assigned "datetime" object
# Remember to localize it with the pytz localize tool, otherwise it will be in the wrong time zone!
obs_time_notz2 = datetime.datetime(year=2020, month=10, day=6, hour=20, minute=44, second=35)
obs_time2 = tz.localize(obs_time_notz)
```

In [25]:
```
# Step 3: Create the target Alt/Az coordinate object
alt2 = 58*u.deg
az2 = 220*u.deg
target_altaz2 = coordinates.SkyCoord(alt=alt2,
                                    az=az2,
                                    location=observer2.location,
                                    obstime=obs_time2, frame='altaz')
```

In [26]:
```
# Step 4: convert the target altaz to galactic
target_galactic2 = target_altaz2.galactic
```

In [27]:
```
# Step 5: Create your topocentric velocity variable
observed_line_of_sight_velocity2 = 27*u.km/u.s
```

In [28]:
```
# Step 6: Convert from topocentric to barycentric
velocity_bary2 = observed_line_of_sight_velocity2 + target_galactic2.radial_velocity_correction()
velocity_bary2
```

Out[28]: $0.17213266\ \frac{\mathrm{km}}{\mathrm{s}}$

In [29]:
```
# Step 7: Convert from barycentric to LSR
# (you'll need to do the special steps noted above - you need "dummy" proper motion, distance)
skycoord_withvel2 = coordinates.SkyCoord(ra=target_altaz2.icrs.ra, dec=target_altaz2.icrs.dec,
                                        radial_velocity=velocity_bary2,
                                        pm_ra_cosdec=0*u.arcmin/u.yr, pm_dec=0*u.arcmin/u.yr, distance=1*u.pc,
                                        frame='icrs')
skycoord_withvel2
```

Out[29]:
```
<SkyCoord (ICRS): (ra, dec, distance) in (deg, deg, pc)
    (286.17169258, 3.74588686, 1.)
 (pm_ra_cosdec, pm_dec, radial_velocity) in (mas / yr, mas / yr, km / s)
    (0., 0., 0.17213266)>
```

In [30]:
```
lsrcoord2 = skycoord_withvel2.transform_to(coordinates.LSR)
lsrcoord2.radial_velocity
```

Out[30]: $16.283863\ \frac{\mathrm{km}}{\mathrm{s}}$

In [31]:
```
target_galactic2.l
```

Out[31]: $37^\circ 46' 35.35305447''$

In [33]:
```
target_galactic2.b
```

Out[33]: $-1^\circ 13' 59.31733987''$

In [34]:
```
vlsr2 = lsrcoord2.radial_velocity
```

In [35]:
```
# Step 8: Compute the Galactocentric radius you're observing
# (we are assuming that we're observing the tangent point along our specified line-of-sight)
rgal2 = np.sin(target_galactic2.l) * r0
rgal2
```

Out[35]: $5.2069533\ \mathrm{kpc}$

In [36]:
```
# Step 9: Compute the mass interior to this orbit
mass2 = rgal2 * vlsr2**2 / constants.G
```

In [37]:
```
mass2.to(u.M_sun)
```

Out[37]: $3.2102399 \times 10^{8}\ \mathrm{M_{\odot}}$