Galactic Astronomy: Problem Set 1

Jonas Powell, *Wesleyan University*

February 11, 2019

Due: Thursday, Feb. 7 by midnight. Late papers are not accepted. If you cannot complete the assignment, hand in what you have completed before the deadline. Consider the deadline to be like the boarding time for an airplane, or the deadline for a grant submission to NASA or NSF. If you miss the deadline, you do not get on the airplane, no matter how good your excuse is. If you miss an NSF or NASA deadline, you do not get the grant, no matter how good your project is. The best advice is ... finish early. You can submit multiple times, right up to the deadline. Whatever your latest submission is, when the deadline occurs, is what will be graded.

Problem 1. Convert the commonly used speed unit of km s^{-1} into a useful unit system for galactic astronomy, pc My^{-1}. That is, 1 kilometer per second equals how many parsecs per million years?

Answer 1. For this problem, we could pretty easily just Google the answer, but it's more fun to work it out unit by unit instead. We begin with the distances.

We would like to convert parsecs to km. We know that there are 206,265 AU/PC, and that 1 AU = 150 million km. Therefore,

$$1 \text{ parsec} = 206265 \times 1.5 \times 10^8 \text{ km}$$

For the time component, we recall that 1 My = 10^6 years, and that a year is made up of 60 seconds/minute, 60 minutes/hour, 24 hours/day, and 365 days/year (more or less). The product of this series is 31,536,000 seconds/year. Therefore, the time component is given by

$$\begin{aligned} 1 \text{ My} &= 10^6 \times (3.1536 \times 10^7) \\ &= 3.1536 \times 10^{13} \text{ seconds} \end{aligned}$$

Putting it all together, we find:

$$\begin{aligned} 1 \text{ pc My}^{-1} &= \frac{206265 \times 1.5 \times 10^8 \text{ km}}{3.1536 \times 10^{13} \text{ seconds}} \\ &\approx 0.98 \text{ km s}^{-1} \\ \rightarrow 1 \text{ km s}^{-1} &= 1.02 \text{ pc My}^{-1} \end{aligned}$$

This is pretty wild! It's saying that a My and parsec have almost exactly the same relationship as the kilometer and second. That's neat.

A note on uncertainties: I am assuming that a year is made up of nice round, consistent number of hours/minutes/seconds, neglecting the reality of leap years/seconds/etc. However, I feel alright doing this because an My is so much bigger than the scale of those fluctuations that any errors in our astronomical velocity measurement are likely far more consequential than these errors.

Problem 2. Calculate the value of the constant in the equation:

$$v_t = \mathrm{const}\mu d$$

where v_t has the units km s^{-1}, μ is in arc-seconds y^{-1} and d is in pc.

Answer 2. To solve this, we may represent each variable by its units, letting k represent our constant, and solve.

$$\begin{aligned}
[\mathrm{km/s}] &= k \times [\mathrm{arcsec/year}] \times [\mathrm{pc}] \\
k &= [\mathrm{km\ year}]/[\mathrm{s\ arcsec\ pc}] \\
&= [\frac{\mathrm{year}}{\mathrm{s}}][\frac{\mathrm{km}}{\mathrm{arcsec\ pc} = \mathrm{AU}}] \\
&= (3.17 \times 10^{-8})(1.5 \times 10^{8}) \\
&= 4.755
\end{aligned}$$

We find a final value for our conversion constant of $k = 4.755$.

Problem 3. Write a computer program that transforms a given RA, Dec (J2000) to galactic coordinates (l,b). Test your program against a Web site that does the same transformation. Be sure to test it for both positive and negative declinations. [Hint: Make sure it works in the tricky region between 0 degrees and -1 degrees.,, such as RA = 04:37:48, Dec = -0:21:25. Also be sure it gives the right answer near the galactic center, l=0.]

Answer 3. Please find the relevant code attached at the end of this file in the function `radec_to_galactic()`. Testing done on the given example, Vega, and a couple others showed functionality.

Problem 4. A fictional G2V star has the following known properties:

Position = 04:24:46.0, +12:37:22.0 (J2000)

Parallax (p) = 0.025″

Proper motion = -5.0 mas yr^{-1} (RA), +24.0 mas yr^{-1} (Dec)

Heliocentric radial velocity (v_r) = +28.0 km/s

Determine the following information for this star. Use appropriate units for galactic astronomy.

Galactic Coordinates (l,b)

Distance (d)

Magnitude of the proper motion vector (μ)

Position Angle of the proper motion vector (PA)

Transverse velocity (v_t)

Speed relative to the Sun (v_{space})

What constellation is this star in?

Is it closer to or further from the center of the Galaxy than the Sun?

Answer 4. Relevant values are given in the table below. Please find my calculations for these values in the Star class in the code attached at the end of this problem set.

	Name	Value	units
0	Stellar Type	G2V	N/A
1	Distance	40	parsec
2	Parallax	0.025	arcsecs
3	Position	(04h24m46.0s, 12d37m22.0s)	[hms, dms]
4	Galactic Coordinates	[182.4771, -24.7694]	degrees
5	Proper Motion (RA)	-2.1209	mas/year
6	Proper Motion (Dec)	24	mas/year
7	Proper Motion Magnitude	24.0935	mas/year
8	Proper Motion Position Angle	-0.0881	radians
9	Radial Velocity	28	km/s
10	Transverse Velocity	4568.13	km/s
11	Space Velocity	4568.22	km/s
12	Host Constellation	Taurus	N/A
13	Distance from Galactic Center	8085.92	parsecs
14	Closer than Sun to GC?	False	boolean

Problem 5. The Gaia Mission is able to measure positional accuracy in the best cases to about 10 μas (micro arc-seconds)!

a) Give an example of what 10 μas is in terms of everyday experience, i.e. something you could a general audience with, to explain to them how good this instrument is at resolving angles.

b) Given its ability to detect proper motions of about 10 μas y^{-1} would Gaia be capable of observing grass growing on the Moon, (assuming grass could grow on the Moon)? If not, from what distance could it observe grass growing? (Be sure to state your assumptions.)

c) Could Gaia detect the rotation of the Andromeda galaxy? (Again, be sure to state any assumptions you make.)

d) Is Gaia capable of detecting the proper motion of a typical quasar? (.... assumptions...)

Answer 5. 10 μas is a really small number - $10 \times 10^{-6}[\frac{\mathrm{as}}{\mu\mathrm{as}}] \times 206265^{-1}[\frac{\mathrm{radians}}{\mathrm{as}}] \approx 5 \times 10^{-11}$ radians. To get a feel for what that means, we can use the small angle approximation for tangent and say $\tan\theta \approx \theta = 5 \times 10^{-11} = \mathrm{O/A}$, so let's think about finding something that is eleven orders of magnitude - again, ELEVEN ORDERS OF MAGNITUDE - smaller than something else.

Part A: Let's say our smaller object is 1mm in scale; in this case, we should be looking for a length that is of order 10^8 meters, or hundreds of thousands of kilometers (to reclaim that coefficient of 5 that was in our conversion, let's say we are looking at length scales of around 500,000 km).

Let's first identify a potential source. A quick Google search of ″millimeter sized things″ informed me that, while a penny is a little too big (it's tens of millimeters across), it seems that, in the "United States of America" lettering at the top of the non-Abe Lincoln side of the penny, each letter looks to be of order 1.5x1mm.

Now that we have our source object, let's figure out how far away Gaia could be while still resolving that source. Conveniently enough (leading into the next part of this question), the moon is a little under 400,000 km away from Earth. While that isn't *quite* far enough to actually stretch our resolution to its limit, it still is pretty darn far.

Now we can put that all together into one sentence: If Gaia were on Earth[1], and if Neil Armstrong had accidentially dropped a penny, face down, when he hopped out onto the Moon, we could resolve the lettering in ″United States of America″ on that little tiny piece of copper. That's pretty spectacular.

ADDENDUM: It has come to my attention that a lot of people in the class chose the penny-on-the-moon example. While I now wish I had been a little more original, I'm still so blown away by the scenario that I'm just keeping it anyways, originality be damned.

Part B: According to TheGrassPeople.com[2], grass grows at a pace of around 2-3 cm/week (this feels like too large a number, but since these are order-of-magnitude calculations, it's fine), which we'll call 1 cm/week $\rightarrow$ 50 cm/year. This number is well above our detection limit of 5×10^{-11} radians $\mathrm{year}^{-1} \approx 1$ mm year^{-1}.

Part C: According to the first plot of Andromeda's rotation curve that I plucked from Google Images[3], Andromeda rotates at around 200 km/s (obviously with radial variations, but let's ignore those for now) and is around 2.5 million light years away. We may convert this to km/year by multiplying by 3×10^7, the approximate number of seconds in a year, yielding 6×10^9 km/year.

At its distance of around 2.5 million light years $\approx 2.4 \times 10^{19}$ km from us, we may find that Andromeda rotates at

[1]...and if it was callibrated to be able to focus on the Moon, and if there didn't exist line-of-sight issues like atmosphere, etc etc.

[2]https://thegrasspeople.com/how-long-grass-grow

[3]https://ned.ipac.caltech.edu/level5/Sept16/Bertone/Figures/figure3.jpg

$$
\begin{aligned}
\frac{v}{d} &= \frac{6 \times 10^9}{2.4 \times 10^{19}} \\
&\approx 2.5 \times 10^{-10} \text{radians/year} \\
&\approx 0.5\,\mu\text{as year}^{-1}
\end{aligned}
$$

With an angular resolution of about 10 μas year^{-1}, this means that we are about an order of magnitude away from being able to actually detect the rotation of the Andromeda Galaxy.

Part D: Since quasars typically are held to be objects with no proper motion, then Gaia would of course be unable to detect any motion. There are a number of publications that discuss the fact that many of the quasars in the Gaia database do, in fact, show proper motion, but since these are atypical, they are not terribly relevant to this question.

Problem 6. With the precision that the Gaia Mission reaches for many stars, a level of about 60 μas per measurement, could it detect the wobble of the Sun due to Jupiter from a distance of 10 pc, assuming it made enough observations over a sufficiently long period of time?

Answer 6. We must begin this problem by assessing how far apart the center of mass and the stellar core are in a system comprised of a Sun-mass star and a Jupiter-mass/major axis planet. Using the center-of-mass equation, we may do this relatively easily. Setting the origin of this calculation to be the star's center and drawing on known values for a Solar mass, Jovian mass, and Jupiter's semi-major axis, we find:

$$
\begin{aligned}
x_{\text{CoM}} &= \frac{m_\odot x_\odot + m_{\text{Jup}} x_{\text{Jup}}}{m_\odot + m_{\text{Jup}}} \\
&= \frac{(1.9 \times 10^{27}\ \text{kg})(7.78 \times 10^{11}\text{m})}{1.9 \times 10^{27}\ \text{kg} + 2 \times 10^{30}\ \text{kg}} \\
&= 7.3 \times 10^8\ \text{meters}
\end{aligned}
$$

This means that the barycenter of this system would be a mere 70,000 km away from the star's center.

We may now find the parallax angle that would be caused by such a motion by taking the ratio of twice that distance (as it swings from side to side over the course of a year) over the 10 parsecs that we are observing from.

$$
\begin{aligned}
p &= \frac{1.46 \times 10^9\ \text{meters}}{10\ \text{parsecs} = 3 \times 10^{17}\ \text{meters}} \\
&= 4.8 \times 10^{-9}\ \text{radians} \\
&= 10^{-3}\ \text{arcsec}
\end{aligned}
$$

This is obviously significantly more than the 60μas per measurement, meaning that it would be relatively easy to detect the wobble of the Sun caused by Jupiter from 10pc with Gaia.

```python
import numpy as np
import pandas as pd
import astropy.units as u
from inspect import *
from astropy.coordinates import SkyCoord, Angle, get_constellation

d_sun_GC = 8122     # Sun/GC distance (parsec)



# Problem 3
def radec_to_galactic_astropy(coords):
    """
    Convert RA/dec coordinates to galactic (l, b) coordinates.

    Args: coords (tuple of strs): RA, dec values in a format understood
                                  by astropy.coordinates.Angle
    Returns: (l, b) tuple, in degrees.
    """
    ra_hms, dec_hms = Angle(coords[0]), Angle(coords[1])
    radec_coords_deg = SkyCoord(ra=ra_hms, dec=dec_hms, frame='icrs')
    galactic_coords_str = radec_coords_deg.transform_to('galactic').to_string()
    galactic_coords_degs = [float(coord) for coord in galactic_coords_str.split(' ')]
    return galactic_coords_degs


def radec_to_galactic(coords):
    """
    Convert RA/dec coordinates to galactic (l, b) coordinates by hand.

    Sources:
        NGP coords taken from:
        https://en.wikipedia.org/wiki/Galactic_coordinate_system

        Conversion formula adapted from:
        http://www.atnf.csiro.au/people/Tobias.Westmeier/tools_coords.php
    Args:
        coords (tuple of strs): RA, dec values in a format understood
                                       by astropy.coordinates.Angle
    Returns: (l, b) tuple, in degrees.
    """

    def gross_coords_to_rads(coords):
        ra, dec = coords
        coords = SkyCoord(ra=ra, dec=dec, frame='icrs')
        ra_rad, dec_rad = [float(a) * np.pi/180
                           for a in coords.to_string().split()]
        return (ra_rad, dec_rad)

    ra, dec = gross_coords_to_rads(coords)
    ra_NGP, dec_NGP = gross_coords_to_rads(['12h51m26.00s', '+27d 7m 42.0s'])
    l_NCP = 122.93 * np.pi/180

```

```python
    b = np.arcsin(np.sin(dec_NGP) * np.sin(dec) \
                  + np.cos(dec_NGP) * np.cos(dec) \
                  * np.cos(ra - ra_NGP))

    x1 = np.cos(dec) * np.sin(ra - ra_NGP)
    x2 = np.cos(dec_NGP) * np.sin(dec) \
         - np.sin(dec_NGP) * np.cos(dec) * np.cos(ra - ra_NGP)

    # Arctan2 is basically a smart version of arctan(x1/x2)
    l = l_NCP - np.arctan2(x1, x2)

    # Convert to degrees and round out to 4 decs for prettiness.
    l, b = round(l * 180/np.pi, 4), round(b * 180/np.pi, 4)
    return [l, b]

# Test the handwritten converter against astropy's answers
vega_coords = ['18h 36m 56s', '+38d47m1s']
test_coords = ['04h37m48s', '-0d21m25s']
print radec_to_galactic(vega_coords)
print radec_to_galactic_astropy(vega_coords)




# Problem 4
def distance_to_galactic_center(galactic_coords, d):
    """
    Find the distance from the Galactic Center of an object at l, b, d.

    Args:
        galactic_coords (tuple of floats): l and b angles, in degrees.
        d (float) distance from Sun to star, in parsecs.
    """
    l, b = galactic_coords[0] * 3600, galactic_coords[1] * 3600
    h_star_gcp, d_star_sun = d * np.sin(b), d * np.cos(b)
    d_star_gc = np.sqrt(d_star_sun**2 + d_sun_GC**2 - 2*d_star_sun*d_sun_GC*np.cos(l))
    return d_star_gc

vega = radec_to_galactic_astropy(vega_coords)


class Star:
    """
    A Python class containing information about a fictional star defined
    by a given parameter set. Attributes include:

    - init_params: the initial list of parameters provided.
    - galactic_coords: the source's galactic coordinates.
    - pm_mag: the magnitude of the proper motion vector.
    - pm_posang: the position angle of the proper motion vector.
    - v_transverse: the source's transverse velocity.
    - v_space: the source's velocity relative to the Sun.
    - constellation: the constellation that the source is in.
```

```python
    - d_from_GC: the source's distance from Galactic Center (parsecs)
    - closer: a boolean telling whether or not the source is
              closer to the Galactic Center than the Sun.

    To check the value of a single parameter (i.e. distance), print Star(params).distance
    To see a dataframe of all the values, pass print_df=True.


    """

    def __init__(self, params, print_df=True, print_help=False):
        """
        Get some info about a star from some other info.

        Args (unpacked from params):
            stellar_type (str): What kind of star it is (i.e. G2V)
            position (tuple of strs): RA, dec values in a format understood
                                      by astropy.coordinates.Angle
            parallax (float): parallax angle, in arcsecs
            proper_motion (tuple of floats): proper motion in RA/dec, in mas/year.
            rv (float): radial velocity, in km/s
        """
        stellar_type, position, parallax, proper_motion, v_radial = params
        self.init_params = params
        self.stellar_type = stellar_type
        self.proper_motion = proper_motion      # [mas/year, mas/year]
        self.distance = 1/parallax              # parsecs
        self.parallax = parallax                # arcsecs
        self.position = position                # [hms, dms]
        self.v_radial = v_radial                # km/s

        self.galactic_coords = radec_to_galactic(self.position)  # degrees

        # Proper motion, described in Cartesian components
        self.pm_dec = self.proper_motion[1]
        # We don't need to scale by cos(dec) because the units are already in mas/year
        self.pm_ra = self.proper_motion[0] #* np.cos(self.pm_dec)

        # Proper motion, described in angular components
        self.pm_mag = np.sqrt(self.pm_ra**2 + self.pm_dec**2)        # mas/year
        # PA = angle east of north
        self.pm_posang = round(np.arctan(self.pm_ra/self.pm_dec), 4) # radians

        self.v_transverse = 4.74 * self.pm_mag * self.distance       # km/s

        # Space velocity is the third leg of the v_trans/v_rad triangle.
        self.v_space = np.sqrt(self.v_transverse**2 + self.v_radial**2)

        star_obj = SkyCoord(Angle(position[0]), Angle(position[1]), frame='icrs')
        self.constellation = get_constellation(star_obj)

        self.d_from_GC = self.distance_to_galactic_center()         # parsecs
        self.closer = True if self.d_from_GC > d_sun_GC else False
```

```

        d = [{'Name': 'Stellar Type', 'Value': self.stellar_type, 'units': 'N/A'},
             {'Name': 'Distance', 'Value': self.distance, 'units': 'parsec'},
             {'Name': 'Parallax', 'Value': self.parallax, 'units': 'arcsecs'},
             {'Name': 'Position', 'Value': self.position, 'units': '[hms, dms]'},
             {'Name': 'Galactic Coordinates', 'Value': self.galactic_coords,
              'units': 'degrees'},
             {'Name': 'Proper Motion (RA)', 'Value': self.pm_ra, 'units': 'mas/year'},
             {'Name': 'Proper Motion (Dec)', 'Value': self.pm_dec, 'units': 'mas/year'},
             {'Name': 'Proper Motion Magnitude', 'Value': self.pm_mag, 'units': 'mas/year'
    },
             {'Name': 'Proper Motion Position Angle', 'Value': self.pm_posang,
              'units': 'radians'},
             {'Name': 'Radial Velocity', 'Value': self.v_radial, 'units': 'km/s'},
             {'Name': 'Transverse Velocity', 'Value': self.v_transverse, 'units': 'km/s'},
             {'Name': 'Space Velocity', 'Value': self.v_space, 'units': 'km/s'},
             {'Name': 'Host Constellation', 'Value': self.constellation, 'units': 'N/A'},
             {'Name': 'Distance from Galactic Center', 'Value': self.d_from_GC,
              'units': 'parsecs'},
             {'Name': 'Closer than Sun to GC?', 'Value': self.closer, 'units': 'N/A'}
             ]

        self.full_param_df = pd.DataFrame(d)

        if print_help:
            print getdoc(self), '\n\n'

        if print_df:
            print self.full_param_df

    def distance_to_galactic_center(self):
        """
        Find the distance from the Galactic Center of an object at l, b, d.

        Args:
            galactic_coords (tuple of floats): l and b angles, in degrees.
            d (float) distance from Sun to star, in parsecs.
        """
        l, b = self.galactic_coords
        h_star_gcp = self.distance * np.sin(b)
        d_star_sun = self.distance * np.cos(b)
        d_star_gc = np.sqrt(d_star_sun**2 + d_sun_GC**2 - 2*d_star_sun*d_sun_GC*np.cos(l))
        return d_star_gc

# params = [stellar_type, position, parallax, proper_motion, rv]
params = ['G2V', ('04h24m46.0s', '12d37m22.0s'), 0.025, (-5.0, 24.0), 28.0]
prob4_star = Star(params, print_df=True, print_help=True)




# Problem 6
def sun_jup_com():
```

```
    # All units MKS
    m_jup, m_sol = 1.9e27, 2e30
    sma_jup = 7.78e11

    com = (m_jup * sma_jup)/(m_jup + m_sol)
    return com
```