PS2

September 22, 2023

1 ASTR 1040 Problem Set 2

For reference, all of the `astropy` constants as well as examples can be found here. Likewise all of the units and their names / how to access them can be found here.

To create a variable with units you need to multiply by the corresponding unit class:

`d = 1 * units.au`

You can then convert unitful quantities to other units by calling the `to` method and passing the desired end unit class:

`d_m = d.to(units.m)`

If you have a ratio of quanities where all the units should cancel out, you can obtain the dimensionles number using the `dimesionless_unscaled` method. For example writing:

`d_m/d`

yields $1.4959787 \times 10^{11} \frac{\mathrm{m}}{\mathrm{AU}}$, but doing:

`(d_m/d).to(units.dimensionless_unscaled)`

returns `1.0` as expected.

If you get weird units at the end but you think your answer should be right, you can always try converting them to SI units with the `.si` method (i.e. `d.si`).

For a more general Python cheatsheet, check the top of your PS1 notebook.

1.0.1 Don't forget to run the cell below before starting your homework!!!

```
[ ]: #SETUP CELL (modify at your own peril)
     from astropy import units #access units by doing units.<unit> (i.e. units.au)
     from astropy import constants
     import numpy as np #common math functions (i.e. np.sin(x)) and better arrays (i.
       ↪e. np.array([1,2,3])
     import matplotlib.pyplot as plt #plotting functions (i.e. plt.plot(x,y))
     G = constants.G # gravitational constant
     M_sun = constants.M_sun # mass of the sun
     R_sun = constants.R_sun # radius of the sun
     L_sun = constants.L_sun # luminosity of the sun
     M_earth = constants.M_earth # mass of the earth
```

```
R_earth = constants.R_earth # radius of the earth
M_jup = constants.M_jup # mass of jupiter
R_jup = constants.R_jup # radius of jupiter
sigma_sb = constants.sigma_sb # Stefan-Boltzmann constant
c = constants.c # speed of light
h = constants.h # Planck constant
k_B = constants.k_B # Boltzmann constant
m_e = constants.m_e # mass of electron
m_p = constants.m_p # mass of proton
m_n = constants.m_n # mass of neutron (basically just the mass of a proton but␣
 ↪whatever)
g0 = constants.g0 # standard gravity, 9.8 m/s^2
e = constants.e # absolute value of electron/proton charge
```

1.1 Q1 (20 points)

1.1.1 Properties of stars

Consider the stellar data in the following table (run the code cell below to see the table!):

```
[ ]: import pandas as pd
     d = {
         "Star": ["Aldebaran","Alpha Centauri␣
      ↪A","Antares","Canopus","Fomalhaut","Regulus","Sirius","Spica"],
         "Absolute magnitude": [-0.2,4.4,-4.5,-3.1,2.0,-0.6,1.4,-3.6],
         "Apparent magnitude": [0.9,0.0,0.9,-0.7,1.2,1.4,-1.4,0.9],
         "Spectral type": ["K5","G2","M1","F0","A3","B7","A1","B1"],
         "Luminosity class": ["III","V","I","II","V","V","V","V"]
     }
     data = pd.DataFrame(data=d)
     data
```

```
[ ]:                Star  Absolute magnitude  Apparent magnitude Spectral type  \
     0          Aldebaran                -0.2                 0.9            K5
     1  Alpha Centauri A                  4.4                 0.0            G2
     2            Antares                -4.5                 0.9            M1
     3            Canopus                -3.1                -0.7            F0
     4          Fomalhaut                 2.0                 1.2            A3
     5            Regulus                -0.6                 1.4            B7
     6             Sirius                 1.4                -1.4            A1
     7              Spica                -3.6                 0.9            B1

       Luminosity class
     0              III
     1                V
     2                I
     3               II
     4                V
```

```
5                V
6                V
7                V
```

This is a pandas DataFrame, a common tool used in the sciences / industry to store and display data in Python. It has many useful features which you won't really need to use, but one really nice feature that may be helpful for you in answering this first question is that you can sort DataFrames very easily. For example, if we wanted to re-sort the DataFrame above in terms of absolute magnitude, we could run the following command:

[ ]:
```
data.sort_values(by=['Absolute magnitude'])
```

[ ]:
```
                Star  Absolute magnitude  Apparent magnitude Spectral type  \
2            Antares                -4.5                 0.9            M1
7              Spica                -3.6                 0.9            B1
3            Canopus                -3.1                -0.7            F0
5            Regulus                -0.6                 1.4            B7
0          Aldebaran                -0.2                 0.9            K5
6             Sirius                 1.4                -1.4            A1
4          Fomalhaut                 2.0                 1.2            A3
1  Alpha Centauri A                  4.4                 0.0            G2

  Luminosity class
2                I
7                V
3               II
5                V
0              III
6                V
4                V
1                V
```

Now you can see that everything is sorted in terms of absolute magnitude, running from smallest to largest. Full documentation for the sort command is available here.

1.1.2 a.) Which star appears brightest and which appears faintest in our sky? Answer using the Markdown and/or code cells below:

[ ]:
```
data.sort_values(by=['Apparent magnitude'])
```

[ ]:
```
                Star  Absolute magnitude  Apparent magnitude Spectral type  \
6             Sirius                 1.4                -1.4            A1
3            Canopus                -3.1                -0.7            F0
1  Alpha Centauri A                  4.4                 0.0            G2
0          Aldebaran                -0.2                 0.9            K5
2            Antares                -4.5                 0.9            M1
7              Spica                -3.6                 0.9            B1
4          Fomalhaut                 2.0                 1.2            A3
```

```
5            Regulus                -0.6                 1.4           B7

  Luminosity class
6                V
3               II
1                V
0              III
2                I
7                V
4                V
5                V
```

- Absolute Magnitude: how bright a star would be if seen from a standard distance
- Apparent Magnitude: how bright a star is from Earth

The star that appears the brightest in our sky means the star with the lowest apparent magnitude, which is **Sirius**.

1.1.3 b.) Which star has the greatest luminosity and which has the lowest? Answer using the Markdown and/or code cells below:

[ ]:
```
data.sort_values(by=['Luminosity class'])
```

[ ]:
```
               Star  Absolute magnitude  Apparent magnitude Spectral type  \
2           Antares                -4.5                 0.9            M1
3           Canopus                -3.1                -0.7            F0
0         Aldebaran                -0.2                 0.9            K5
1  Alpha Centauri A                 4.4                 0.0            G2
4         Fomalhaut                 2.0                 1.2            A3
5           Regulus                -0.6                 1.4            B7
6            Sirius                 1.4                -1.4            A1
7             Spica                -3.6                 0.9            B1

  Luminosity class
2                I
3               II
0              III
1                V
4                V
5                V
6                V
7                V
```

The star with the greatest luminosity is Antares.

1.1.4 c.) Which star has the highest surface temperature and which has the lowest? Answer using the Markdown and/or code cells below:

```
[ ]: data.sort_values(by=['Luminosity class'])
```

```
[ ]:                Star  Absolute magnitude  Apparent magnitude Spectral type  \
     2           Antares                -4.5                 0.9            M1
     3           Canopus                -3.1                -0.7            F0
     0         Aldebaran                -0.2                 0.9            K5
     1  Alpha Centauri A                 4.4                 0.0            G2
     4         Fomalhaut                 2.0                 1.2            A3
     5           Regulus                -0.6                 1.4            B7
     6            Sirius                 1.4                -1.4            A1
     7             Spica                -3.6                 0.9            B1

       Luminosity class
     2                I
     3               II
     0              III
     1                V
     4                V
     5                V
     6                V
     7                V
```

The star with the highest surface temperature is Regulus, since its spectral type is B7

1.1.5 d.) Which star is most similar to the Sun? Answer using the Markdown and/or code cells below:

The star more similar to our Sun is Alpha Centauri A, which is both a G spectral type and V luminosity class.

1.1.6 e.) Which star is a red supergiant? Answer using the Markdown and/or code cells below:

The red supergiant is Antares, with a luminositiy class of I.

1.1.7 f.) Which star has the largest radius? Answer using the Markdown and/or code cells below:

The star with the largest radius would be Antares, because supergiants are the largest stars.

1.1.8 g.) Which stars have finished burning hydrogen in their cores? Answer using the Markdown and/or code cells below:

The stars that have finished burning H in their cores, implying that nuclear fusion is no longer occurring, are all the stars that are not on the main sequence: - Antares - Canopus - Aldebaran

1.1.9 h.) Among the main-sequence stars listed, which is the most massive? Answer using the Markdown and/or code cells below:

```
[ ]: data.sort_values(by=['Luminosity class'])
```

```
[ ]:                Star  Absolute magnitude  Apparent magnitude Spectral type  \
     2           Antares                -4.5                 0.9            M1
     3           Canopus                -3.1                -0.7            F0
     0         Aldebaran                -0.2                 0.9            K5
     1  Alpha Centauri A                 4.4                 0.0            G2
     4         Fomalhaut                 2.0                 1.2            A3
     5           Regulus                -0.6                 1.4            B7
     6            Sirius                 1.4                -1.4            A1
     7             Spica                -3.6                 0.9            B1

       Luminosity class
     2                I
     3               II
     0              III
     1                V
     4                V
     5                V
     6                V
     7                V
```

Of the main sequence stars listed, the most massive is Antares, because red giants are evolved from massive stars.

1.1.10 i.) Among the main-sequence stars listed, which has the longest lifetime? Answer using the Markdown and/or code cells below:

The stars with the longest lifetimes will be in the bottom right corner of the HR diagram on the main sequence. Therefore, Alpha Centauri A is likely the one with the longest lifetime.

1.2 Q2 (60 points)

1.2.1 Calculating the physical properties of stars from observables: Procyon

Procyon is the 7th brightest star in the sky. From telescope observations, we know that the wavelength of maximimum intensity, λ_{max}, is about 444 nm and the total intensity (apparent brightness) is $1.81\times10^{-8}\ \frac{\text{W}}{\text{m}^2}$. The measured parralax angle for Procyon is 285 mas (milliarcseconds).

1.2.2 a.) How far away is Procyon? Use the code cell below to calculate and provide your answer in lightyears.

```
[ ]: alpha = (0.285 * units.arcsec).to(units.rad)
     x = 1 * units.AU
     d = (x / alpha)*units.rad
     d.to(units.lyr)
```

[ ]: 11.444083 lyr

1.2.3 b.) What is the luminosity of Procyon? Use the code cell below to calculate and express your answer in terms of $L_\odot$.

[ ]:
```
# use d and I
# L = 4*pi*d^2 * I

I = 1.81e-8 * (units.watt / units.m**2)
L = 4 * np.pi * (d**2) * I
L_solar_units = (L / L_sun).si
print(f"{L_solar_units} solar units")
```

```
6.965108405618085 solar units
```

1.2.4 c.) What is the surface temperature of Procyon? Use the code cell below to calculate and provide your answer in K.

[ ]:
```
# Use Wien's Law

 _max = 444 * units.nm
T = (2.9e6 * units.nm * units.K) /  _max
T
```

[ ]: 6531.5315 K

1.2.5 d.) What is the intensity of light at Procyon's surface? Use the code cell below to calculate and provide your answer in $\frac{W}{m^2}$.

[ ]:
```
# flux at surface =>  *T**4 == L / 4*pi*d**2

flux = sigma_sb * T**4
flux.to(units.watt / units.m**2)
```

[ ]: $1.0319813 \times 10^{8}\ \frac{\mathrm{W}}{\mathrm{m}^2}$

1.2.6 e.) What is Procyon's surface area? What is its radius? Use the code cell below to calculate both values, and compare the radius to our Sun's.

[ ]:
```
# use stefan-boltzmann const and T
# I = P/A =  *T**4
# A = L /(P/A)
# R = sqrt(A/4*pi)

A = L / flux
R = np.sqrt(A/4*np.pi)
(print(f"{R.si} is 1 order of magnitude larger than the radius of our Sun's,␣
 ↪{R_sun}."))
```

```
# R_sun
```

```
4504627890.580754 m is 1 order of magnitude larger than the radius of our Sun's,
695700000.0 m.
```

1.2.7 f.) Procyon is half of a binary system. The two stars are separted by 4.3 arcseconds (semimajor axis) and have an orbital period of 40.82 years. What is the combined mass of both stars in the Procyon system? Use the code cell below to calculate and express your answer in terms of solar masses ($M_\odot$).

[ ]:
```
# (M_1 + M_2)*P**2 = a**3
# let combined mass = M: M*P**2 = a**3
# M = a**3 / P**2

a = (4.3 * units.arcsec).to(units.rad)
# x = 1 * units.AU
a_d = (x / a)*units.rad
a_d.to(units.au)
# R_2 = a_d - R

P = 40.82 * units.year
# M = a**3 / P**2
M = a_d**3 / P**2
M.si # couldn't get dimensions to work out :-(

# R_2.to(units.au)
# R
# a_d
```

[ ]: $2.226857 \times 10^{29}\ \frac{\mathrm{m^3}}{\mathrm{s^2}}$

1.3 Q3 (10 points)

What is the most fundamental property of stars? Why? Use the markdown cell below to provide your answer.

The most fundamental property of stars is *mass*. Mass determines the most fundamental properties of stars such as luminosity, temperature, lifespan, and color. We can understand how stars formed, when, and how by categorizing them based on mass. For example, the HR diagram illustrates Temperature and Luminosity of stars, which we can use to infer their mass and thus their stellar evolutions.

1.4 Q4 (10 points)

How do we age date stars? Use the markdown cell below to provide your answer.

Stars are age dated via observing their placement within the HR diagram. Stars that fall along the main sequence are actively fusing hydrogen in their core, and this information can be used to determine the age of star clusters. When stars are bluer in color, we can infer that the age is relatively younger than a star that is redder. If it is redder, that means the blue, low-mass, longer-living stars are gone, meaning more time must have passed.