plots

October 14, 2024

0.1 Assignment 5

[1]:
```
#pip install mesa_reader
```

0.1.1 Question 1: In Simulation.

0.1.2 HR diagram

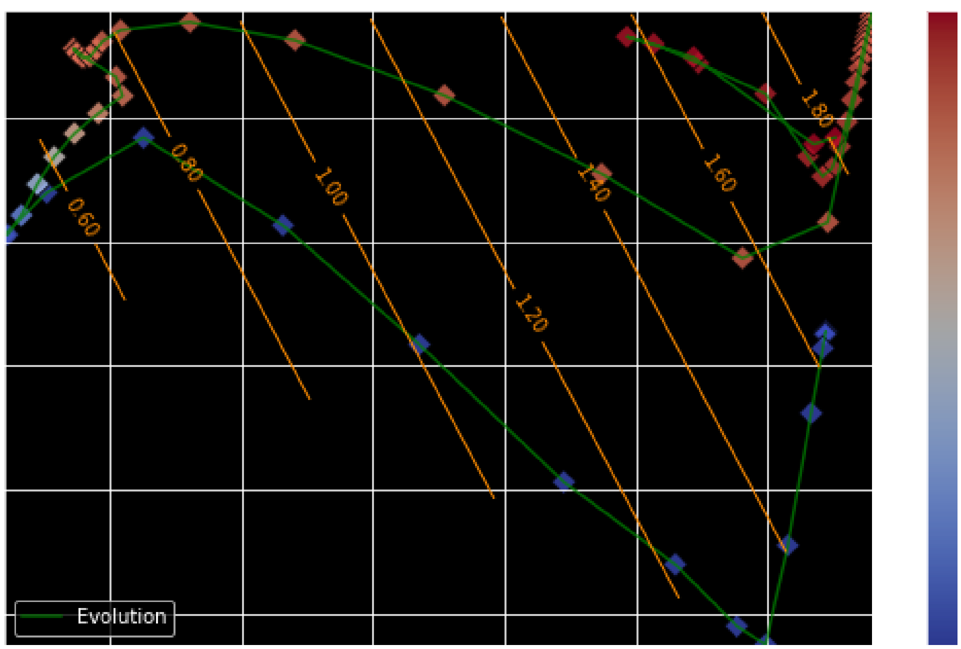
[2]:
```
import mesa_reader as mr
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.backends.backend_pdf import PdfPages
import pandas as pd

plt.style.use('dark_background')
plotpdf = PdfPages("Plots_task5.pdf")
```

[3]:
```
# MesaData object
h = mr.MesaData('./mesa_assgn5/LOGS/history.data')
p = mr.MesaData('./mesa_assgn5/LOGS/profile1.data')
```

[4]:
```
# Extract data for plotting
log_Teff = h.log_Teff                    # Logarithm of the effective temperature
log_L = h.log_L                          # Logarithm of the luminosity
star_age = h.star_age                    # Age of the star in years
star_mass = h.star_mass                  # Mass of the star in solar masses
log_R = h.log_R                          # Logarithm of the stellar radius in solar␣
 ↪units
```

[5]:
```
# Question 2a
# HR Diagram

plt.figure(figsize=(10,6))

sc = plt.scatter(log_Teff, log_L, c=star_age, cmap='coolwarm', marker='D',␣
 ↪edgecolor='k', s=80, alpha=0.75)
plt.plot(log_Teff, log_L, color='green', linewidth=1.5, linestyle='-', alpha=0.
 ↪75, label='Evolution')
```

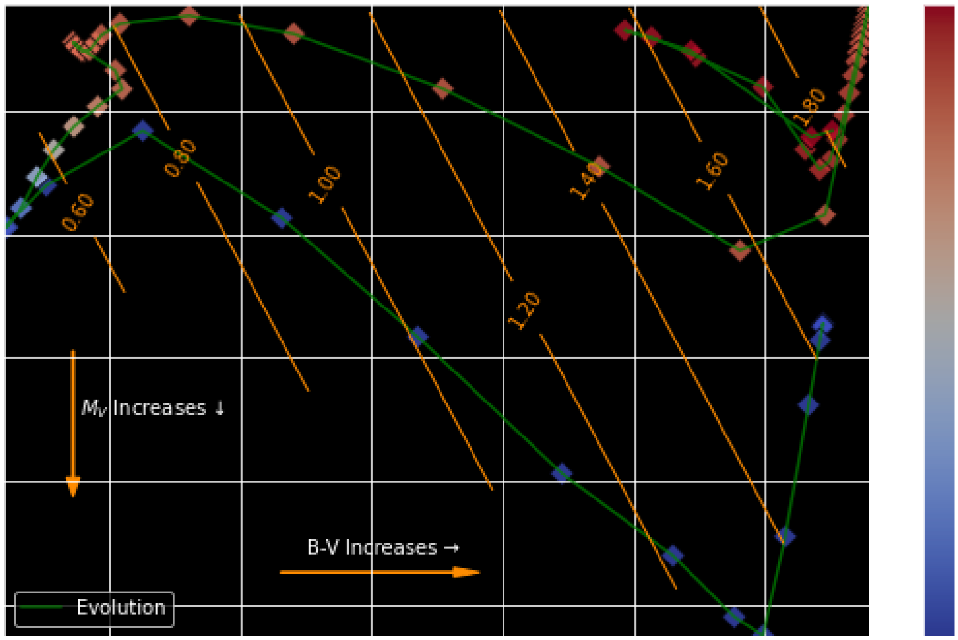
```
plt.gca().invert_xaxis()

contour = plt.tricontour(log_Teff, log_L, log_R, levels=8, linewidths=1,␣
 ↪colors='darkorange')
plt.clabel(contour, inline=True, fontsize=10, fmt='%.2f')

cbar = plt.colorbar(sc)
cbar.set_label('Age (years)', rotation=270, labelpad=15)

plt.title('HR Diagram', fontsize=25, fontweight='bold')
plt.xlabel(r'$\log_{10}(T_{eff} / K)$', fontsize=15)
plt.ylabel(r'$\log_{10}(L / L_\odot)$', fontsize=15)


plt.grid(visible=True)
plt.legend(loc='best')

plotpdf.savefig()
plt.show()
```



HR Diagram showing the evolution of star with $M = 6M_{\odot}$. Color: age of the star. Contours: logarithm of the stellar radius

[6]:
```python
# Question 2.b
#Indicate the direction of increasing values: left for B  V and up for M_V.

log_Teff = h.data('log_Teff')  # Logarithm of effective temperature
log_L = h.data('log_L')  # Logarithm of luminosity
star_age = h.star_age  # Age of the star



# HR Diagram
plt.figure(figsize=(10,6))

sc = plt.scatter(log_Teff, log_L, c=star_age, cmap='coolwarm', marker='D', s=80, edgecolor='k', alpha=0.75)
plt.plot(log_Teff, log_L, color='green', linewidth=1.5, linestyle='-', alpha=0.75, label='Evolution')

contour = plt.tricontour(log_Teff, log_L, log_R, levels=8, linewidths=1, colors='darkorange')
plt.clabel(contour, inline=True, fontsize=10, fmt='%.2f')

cbar = plt.colorbar(sc)
cbar.set_label('log(Age) [Years]', fontsize=12)

plt.xlabel(r'$\log_{10}(T_{eff} / K)$', fontsize=15)
plt.ylabel(r'$\log_{10}(L / L_{\odot})$', fontsize=15)
plt.gca().invert_xaxis()


plt.grid(visible=True)
plt.legend(loc='best')




plt.annotate('B-V Increases ', xy=(0.35, 0.13), xycoords='axes fraction', fontsize=10, color='white')
plt.annotate(r'$M_V$ Increases ', xy=(0.088, 0.35), xycoords='axes fraction', fontsize=10, color='white')

plt.arrow(0.32, 0.1, 0.2, 0, transform=plt.gca().transAxes, color='darkorange', width=0.002, head_width=0.015, head_length=0.03)
plt.arrow(0.08, 0.45, 0, -0.2, transform=plt.gca().transAxes, color='darkorange', width=0.002, head_width=0.015, head_length=0.03)

plt.title(r'H-R Diagram: B-V and $M_V$', fontsize=25, fontweight='bold')
```

```
plotpdf.savefig()
plt.show()
```



The B-V color index measures the color of the star, which is directly related to its temperature. As the star cools, the B-V color index increases; thus, it would increase to the right on the horizontal axis, opposite to the direction of decreasing temperature. The absolute magnitude (M_V) represents the star's intrinsic brightness, where higher magnitude values indicate a dimmer star. Consequently, (M_V) increases downward on the vertical axis, opposite to the increase in luminosity. However, since we could not find the B-V color index values in the history files and found no relevant information in the documentation, we can not plot it.

[7]:
```
# Question 2c
#What is the spectral type of your star on the main sequence?

# The effective temperature at ZAMS
Teff_zams = h.data('log_Teff')[0]
Teff_zams_K = 10**Teff_zams

# Determine the spectral type
if Teff_zams_K > 30000:
    spectral_type_zams = 'O'
elif Teff_zams_K > 10000:
```

```python
    spectral_type_zams = 'B'
elif Teff_zams_K > 7500:
    spectral_type_zams = 'A'
elif Teff_zams_K > 6000:
    spectral_type_zams = 'F'
elif Teff_zams_K > 5200:
    spectral_type_zams = 'G'
elif Teff_zams_K > 3700:
    spectral_type_zams = 'K'
else:
    spectral_type_zams = 'M'

print(f"Spectral type at zero age main sequence: {spectral_type_zams}")
```

```
Spectral type at zero age main sequence: K
```

[8]:
```python
# Question 2d
#What is the spectral type of your star at the onset of helium burning?

# The onset of helium burning

onset_index = np.where(h.data('log_LHe') > -1)[0][0]  # LHe becomes dominant

Teff_he_burn = h.data('log_Teff')[onset_index]
Teff_he_burn_K = 10**Teff_he_burn

if Teff_he_burn_K > 30000:
    spectral_type_he_burn = 'O'
elif Teff_he_burn_K > 10000:
    spectral_type_he_burn = 'B'
elif Teff_he_burn_K > 7500:
    spectral_type_he_burn = 'A'
elif Teff_he_burn_K > 6000:
    spectral_type_he_burn = 'F'
elif Teff_he_burn_K > 5200:
    spectral_type_he_burn = 'G'
elif Teff_he_burn_K > 3700:
    spectral_type_he_burn = 'K'
else:
    spectral_type_he_burn = 'M'

print(f"Spectral type when helium starts burning: {spectral_type_he_burn}")
```

```
Spectral type when helium starts burning: K
```

[9]:
```python
# Question 2e
# How long does it take for your star to reach the helium core burning phase?

# The age at ZAMS and onset of helium burning

age_zams = h.star_age[0]
age_he_burn = h.star_age[onset_index]

# Duration
time_to_he_burn = age_he_burn - age_zams

print(f"Time at helium core burning: {time_to_he_burn:.2e} yrs")
```

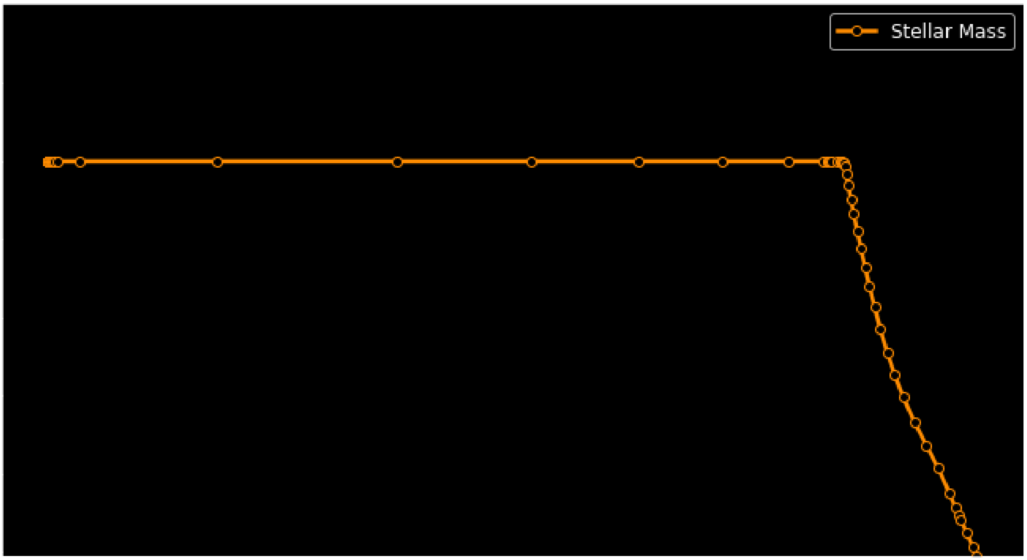
```
Time at helium core burning: 5.62e+07 yrs
```

[10]:
```python
# Question 2f
#Approximately what fraction of this time is spent on the main sequence?

time_main_sequence = time_to_he_burn
total_time = h.star_age[-1]

# fraction of time spent on the main sequence
fraction_main_sequence = time_main_sequence / total_time

print(f"Time fraction spent on main sequence: {fraction_main_sequence:.2%}")
```

```
Time fraction spent on main sequence: 85.82%
```

[11]:
```python
# Question 3
# Mass evolution

#Plot the mass M vs time t for your star. If it is almost constant, reduce the␣
↪range of M until you can see the evolution clearly.

mass = h.data('star_mass')  # Mass of the star
time = h.star_age           # Time in years

plt.figure(figsize=(10, 6))
plt.plot(time, mass, color='darkorange',marker='o',markerfacecolor='black',␣
↪linestyle='-', linewidth=2.5, label='Stellar Mass')

plt.xlabel('Time (Years)', fontsize=15)
plt.ylabel(r'Mass ($M_\odot$)', fontsize=15)
plt.title('Mass Evolution vs Time', fontsize=25, fontweight='bold')
plt.legend(fontsize=12)
plt.tight_layout()
```

```
plt.ylim(min(mass), max(mass)+0.02)
plotpdf.savefig()
plt.show()
```



[12]:
```
plotpdf.close()
```