

## Stellar Evolutionary Models Project

Astronomy 3520

Tim Higginbotham

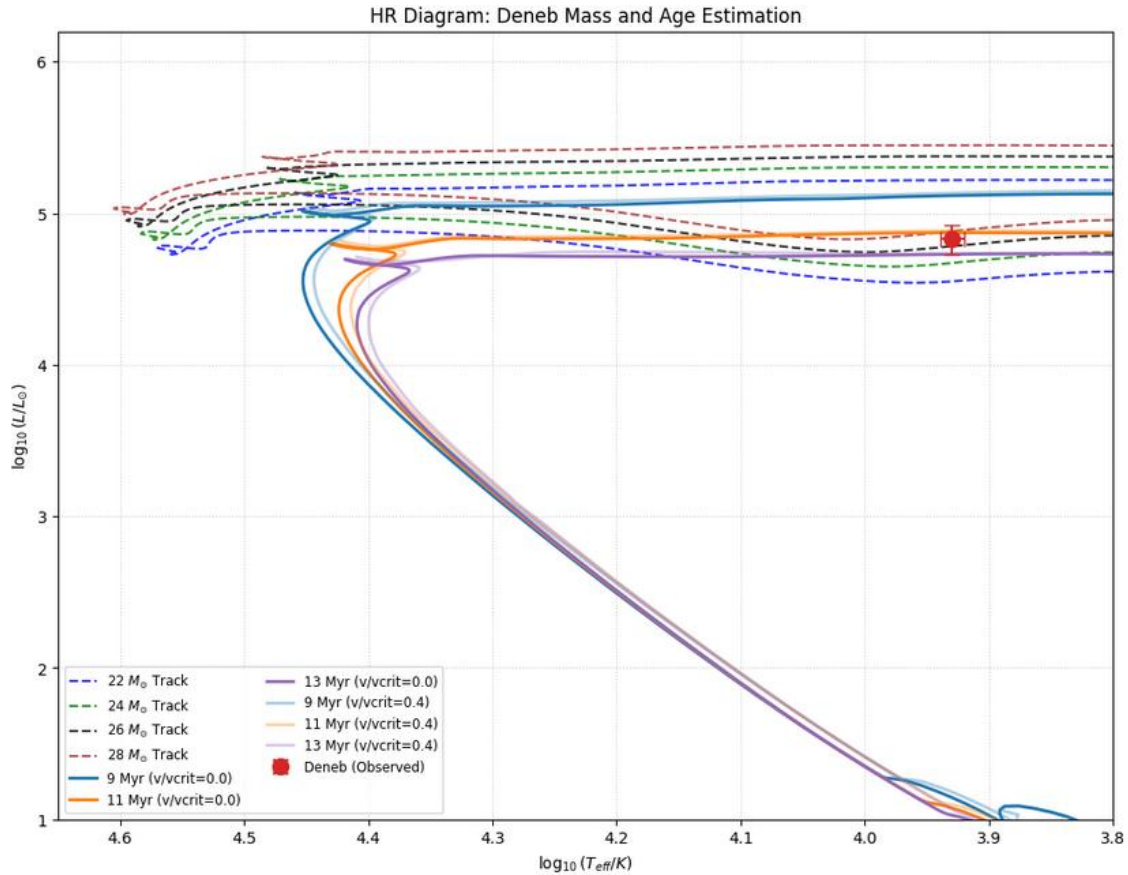
Star: Deneb

Deneb, also known as 'alf Cyg', or Alpha Cygni, is an evolved supergiant star, which means it has left the main sequence, is no longer burning hydrogen in its core, and is much larger than the Sun. It is located in the constellation Cygnus and has an apparent magnitude which varies between 1.21 and 1.29. Its coordinates are 20 hours 41 minutes 25.9 seconds Right Ascension, and +45 degrees, 16 arcminutes, 49 arcseconds declination. Its parallax is 2.31 mas, which places it about 1,430 light-years away from Earth. Its spectral type is A2 Ia, per the Simbad database. (SIMBAD Astronomical Database; Wenger et al., 2000)

At one time it was thought that Deneb was possibly a binary, but more recent observations have found no evidence to support that. The varying nature of the magnitudes may be the result of multiple pulsation periods, and analysis of radial velocities indicates 12 different harmonic pulsation modes. It's possible that Deneb is evolving to a red supergiant stage, and the spectral analysis might be explained by having rotated faster during its main sequence stage. (Deneb, 2025)

### Temperature and Luminosity

The effective temperature of Deneb was adopted as  $T_{\text{eff}} = 8500$ , consistent with its A2 Ia spectral classification. The effective temperature of 8,500 K is adopted from the spectroscopic analysis by Schiller, F., & Przybilla, N. (2008). The uncertainty of 0.01 corresponds to a physical uncertainty of approximately  $\pm 200$  K, consistent with the atmospheric modeling uncertainties described in Schiller & Przybilla (2008). The luminosity was calculated from the observed apparent magnitude  $m_V = 1.25$  and parallax  $p = 2.31$  mas, giving 433 pc. An extinction correction of  $A_V = 0.25$  mag was applied based on the observed color excess, and a bolometric correction of  $BC_V = -0.15$  appropriate for an A-type supergiant was used. These values yield an absolute bolometric magnitude  $M_{\text{bol}} = -7.33$ , corresponding to  $L = 6.8 \times 10^4 L_{\odot}$ . The uncertainty in luminosity ( $\pm 1.5 \times 10^4 L_{\odot}$ ) reflects uncertainties in parallax, extinction, and bolometric correction. All adopted values are explicitly stated so that the calculation can be reproduced.



--- AGE / MASS ESTIMATION ---  
 Best-matching track: 22 Solar Masses  
 Estimated Age: 8.27 Million Years  
 Closest-point (track)  $\log_{10}(T_{\text{eff}}/K) = 3.9316$ ,  $\log_{10}(L/L_{\text{sun}}) = 5.2194$

Deneb plotted on a Hertzsprung–Russell (HR) diagram using the values  $T_{\text{eff}} = 8500$  and  $L = (6.8 \pm 1.5) \times 10^4 L_{\odot}$ . The diagram displays  $\log_{10}(L/L_{\odot})$  versus  $\log_{10}(T_{\text{eff}})$ , with the temperature axis inverted, following HR convention. Overlaid are MIST (MESA Isochrones and Stellar Tracks) evolutionary tracks for 22, 24, 26, and 28  $M_{\odot}$  stars, shown as dashed lines, along with 9, 11, and 13 Myr isochrones for both non-rotating ( $v/v_{\text{crit}} = 0.0$ ) and rotating ( $v/v_{\text{crit}} = 0.4$ ) models. Deneb’s position is marked with error bars reflecting uncertainties in temperature and luminosity. The star lies in the high-luminosity, lower-temperature region characteristic of evolved supergiants, well above the main-sequence portion of the tracks. Its location relative to the evolutionary tracks and isochrones allows an estimate of its mass and age by comparison with the nearest model curves. The mass estimation is a best fit mathematical solution to the way the mass tracks both above and

below the observed location. The plot was generated using python and associated libraries.

### Mass and Age

Based on the HR diagram and comparison with evolutionary models, Deneb is estimated to have a mass of  $22 \pm 3 M_{\odot}$  and an approximate age of 8.27 million years. The program uses a Weighed Euclidean Distance to find the closest match between the observed data and the several evolutionary tracks. The best-matching evolutionary track indicates that Deneb is a high-mass star in a late stage of its evolution. These estimates are subject to uncertainties based on the star's measured luminosity and temperature, represented by the error bars on the observed data point. Visually, Deneb's position overlaps with several mass tracks ranging from approximately 20 to  $28 M_{\odot}$ . Deneb is typically cited with a mass of  $19 \pm 4 M_{\odot}$ . (Schiller & Przybilla (2008)) So, the results are reasonably consistent. Regarding its age, the observed data point sits just below the 9-million-year isochrone; given the proximity to this model and the calculated fit, the age uncertainty is roughly plus or minus one million years. These results show that while the track is the most probable fit, variations in stellar rotation and observational precision allow for a range of possible mass and age values.

### References.

Wenger, M., et al. (2000). The SIMBAD astronomical database. *Astronomy and Astrophysics Supplement Series*, 143, 9–22.

Deneb. (2025). In *Wikipedia*. Retrieved from <https://en.wikipedia.org/wiki/Deneb>

Schiller, F., & Przybilla, N. (2008), 'Quantitative spectroscopy of Deneb', *Astronomy & Astrophysics*, 479(3), 849-858.

### Artificial Intelligence

MathGPT was used to calculate logarithmic age parameters for isochrones and interpret the intersections of evolutionary mass tracks with the observed data. In addition, it was used to synthesize the models into a summary that accounts for the uncertainties in Deneb's luminosity and temperature. It should also be noted that according to A.I., now, there is an updated measurement for parallax of 1.2. I couldn't locate this updated figure for sure, so I used the original number.

## Python Code

```
: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# 1. DENEBA DATA
# Luminosity values are in terms of the sun's Luminosity.

teff_obs = 8500
lum_obs = 68000
lum_err = 15000

log_teff_obs = np.log10(teff_obs)
log_lum_obs = np.log10(lum_obs)
log_lum_low = np.log10(lum_obs - lum_err)
log_lum_high = np.log10(lum_obs + lum_err)

def load_mist_robust(file_path):
    with open(file_path, 'r') as f:
        lines = f.readlines()
        header_line = [l for l in lines if l.startswith('#)][-1]
        col_names = header_line.replace('#', '').split()
    df = pd.read_csv(file_path, sep=r'\s+', comment='#', names=col_names)
    df = df.apply(pd.to_numeric, errors='coerce').dropna(subset=['log_L', 'log_Teff'])
    return df[df['log_L'] > 0]

def load_isochrone(file_path, target_log_age, verbose=True):
    header = None
    with open(file_path, "r") as f:
        for line in f:
            if line.startswith("# EEP"):
                header = line
                break
    if header is None:
        raise ValueError("Could not find '# EEP' header line in the .iso file.")

    col_names = header.lstrip("#").split()
    df = pd.read_csv(file_path, sep=r"\s+", comment="#", names=col_names)
    df = df.apply(pd.to_numeric, errors="coerce")

    if "log_L" not in df.columns and "log_L_Lsun" in df.columns:
        df = df.rename(columns={"log_L_Lsun": "log_L"})

    df = df.dropna(subset=["log10_isochrone_age_yr", "log_Teff", "log_L"])
    ages = np.asarray(df["log10_isochrone_age_yr"].unique(), dtype=float)
    chosen = float(ages[np.argmin(np.abs(ages - target_log_age))])

    if verbose:
        print(f"logAge requested {target_log_age:.3f} → using {chosen:.3f}")

    return df[df["log10_isochrone_age_yr"] == chosen].copy()

try:
    # 1. Load the Mass Tracks
```

```

data19 = load_mist_robust('01900M.track.eep')
data22 = load_mist_robust('02200M.track.eep')
data24 = load_mist_robust('02400M.track.eep')
data26 = load_mist_robust('02600M.track.eep')
data28 = load_mist_robust('02800M.track.eep')

# 2. Load the Isochrones
iso_path_0 = "./MIST_v1.2_feh_p0.00_afe_p0.0_vvcrit0.0_basic.iso"
iso_path_4 = "./MIST_v1.2_feh_p0.00_afe_p0.0_vvcrit0.4_basic.iso"

iso0_13 = load_isochrone(iso_path_0, target_log_age=7.11)
iso0_9 = load_isochrone(iso_path_0, target_log_age=6.95)
iso0_11 = load_isochrone(iso_path_0, target_log_age=7.04)
iso4_13 = load_isochrone(iso_path_4, target_log_age=7.11)
iso4_9 = load_isochrone(iso_path_4, target_log_age=6.95)
iso4_11 = load_isochrone(iso_path_4, target_log_age=7.04)

# 3. Create the Plot
plt.figure(figsize=(12, 9))

# Plot Tracks
# plt.plot(data19['log_Teff'], data19['log_L'], label=r'19 $M_{\odot}$ Track', c
plt.plot(data22['log_Teff'], data22['log_L'], label=r'22 $M_{\odot}$ Track', co
plt.plot(data24['log_Teff'], data24['log_L'], label=r'24 $M_{\odot}$ Track', co
plt.plot(data26['log_Teff'], data26['log_L'], label=r'26 $M_{\odot}$ Track', co
plt.plot(data28['log_Teff'], data28['log_L'], label=r'28 $M_{\odot}$ Track', co

# Plot Isochrones

plt.plot(iso0_9["log_Teff"], iso0_9["log_L"], color="tab:blue", linestyle="-",
plt.plot(iso0_11["log_Teff"], iso0_11["log_L"], color="tab:orange", linestyle="
plt.plot(iso0_13["log_Teff"], iso0_13["log_L"], color='tab:purple', linestyle="

plt.plot(iso4_9["log_Teff"], iso4_9["log_L"], color="tab:blue", linestyle="-",
plt.plot(iso4_11["log_Teff"], iso4_11["log_L"], color="tab:orange", linestyle="
plt.plot(iso4_13["log_Teff"], iso4_13["log_L"], color='tab:purple', linestyle="
# Plot Deneb Observation
plt.errorbar(log_teff_obs, log_lum_obs, xerr=0.01,
             yerr=[[log_lum_obs - log_lum_low], [log_lum_high - log_lum_obs]],
             fmt='o', color='tab:red', label='Deneb (Observed)', capsiz=5, mar

# Standard HR Diagram Settings
plt.xlim(4.65, 3.8) # Automatically inverts axis by putting larger number first
plt.ylim(1.0, 6.2)

plt.xlabel(r'$\log_{10}(T_{\text{eff}} / \text{K})$')
plt.ylabel(r'$\log_{10}(L / L_{\odot})$')
plt.title('HR Diagram: Deneb Mass and Age Estimation')
plt.legend(loc='lower left', fontsize='small', ncol=2)
plt.grid(True, linestyle=':', alpha=0.6)
plt.show()

# 4. CALCULATE AGE RESULTS (Updated to 24 Solar Masses)
tracks = {
    22: data22,
    24: data24,

```

```

    26: data26,
    28: data28
}

# Uncertainties in log-space (used for weighting)
# Luminosity: you already have asymmetric bounds in log-space; use half-range as
sig_logL = 0.5 * ((log_lum_high - log_lum_obs) + (log_lum_obs - log_lum_low))
# Temperature: you used xerr=0.01 on the plot, so use that as sigma in log10(Te)
sig_logT = 0.01

best = None

for M, track in tracks.items():
    # Only consider post-main-sequence ages (your original filter)
    mature = track[track['star_age'] > 5e6].copy()

    # Weighted distance in (logTeff, logL) space
    dist = np.sqrt(((mature['log_L'] - log_lum_obs) / sig_logL)**2 +
                  ((mature['log_Teff'] - log_teff_obs) / sig_logT)**2)

    idx = dist.idxmin()
    age_years = mature.loc[idx, 'star_age']
    dmin = dist.loc[idx]

    if (best is None) or (dmin < best['dmin']):
        best = {
            'M': M,
            'age_years': age_years,
            'dmin': dmin,
            'logL_track': mature.loc[idx, 'log_L'],
            'logT_track': mature.loc[idx, 'log_Teff']
        }

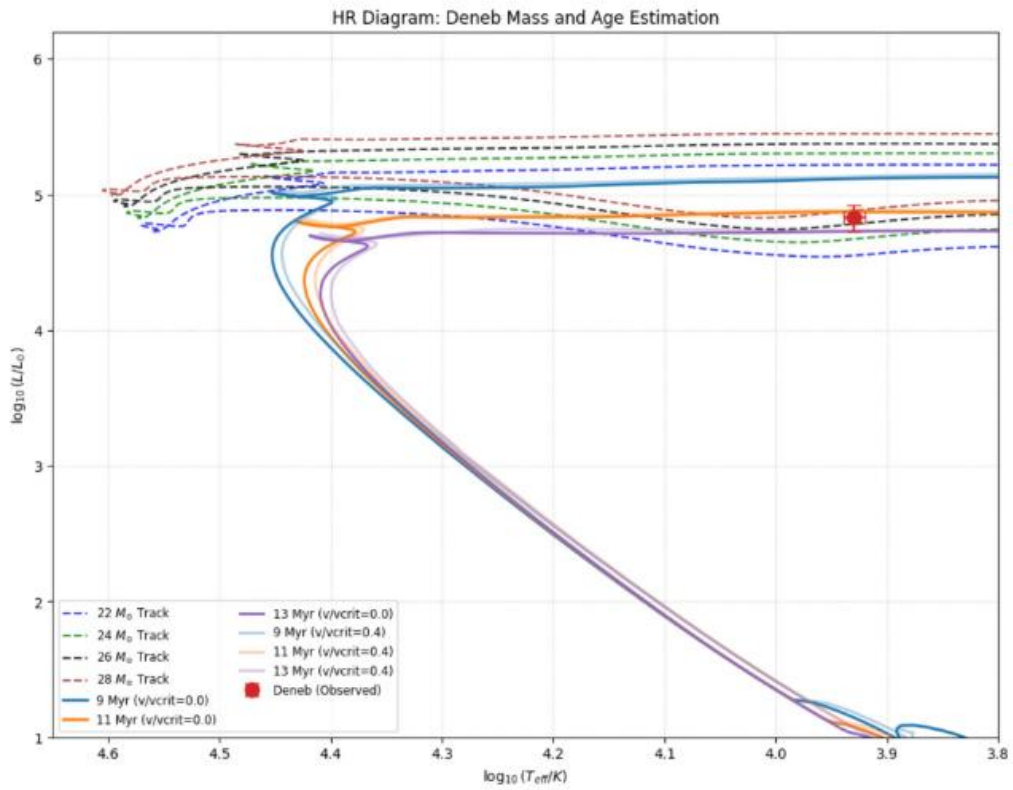
print(f"\n--- AGE / MASS ESTIMATION ---")
print(f"Best-matching track: {best['M']} Solar Masses")
print(f"Estimated Age: {best['age_years']/1e6:.2f} Million Years")
print(f"Closest-point (track) log10(Teff/K) = {best['logT_track']:.4f}, log10(L
except Exception as e:
    print(f"Error: {e}")

```

```

logAge requested 7.110 → using 7.100
logAge requested 6.950 → using 6.950
logAge requested 7.040 → using 7.050
logAge requested 7.110 → using 7.100
logAge requested 6.950 → using 6.950
logAge requested 7.040 → using 7.050

```



--- AGE / MASS ESTIMATION ---

Best-matching track: 22 Solar Masses

Estimated Age: 8.27 Million Years

Closest-point (track)  $\log_{10}(T_{\text{eff}}/K) = 3.9316$ ,  $\log_{10}(L/L_{\text{sun}}) = 5.2194$

TABLE 1.