1 Introduction

–Initial Problem (graph of obs data)
–First thoughts (Hypothesis)
–What Methods are viable?

2 Method

–Which Method did I use and why?
–General approach (structure of main.py?)
–What does Lumiradius.py do?
For this approach I used two programs: main.py and lumiradius.py.

At the core of my main program lies a sequence of nested loops that spans a grid in distance-mass-age. The range of these three variables can be easily adjusted, as can the binning. In accordance to the observational data I try to explain, distance goes from 0 to $r_{max} = 3kpc$. I use a logarithmic binning for mass from $M_{min} = 5M_{\odot}$ to $M_{max} = 50M_{\odot}$. The mass-distribution of my stars will follow the salpeter IMF. Because the IMF is naturally skewed towards smaller stars, I use a logarithmic mass-grid. Age spans from 0 to the maximum age the oldest star in my simulation can reach. Because the main sequence age (t_{ms}) of a star is a strictly monotonic increasing function of M the following is true: $t_{max} = t_{ms}(M_{min})$. With $M_{min} = 5M_{\odot}$ this translates to $t_{max} = 104Myr$. This Axis will also be logarithmic. Even tho the IMF is skewed towards smaller, and thus long lived stars t_{ms} drops faster, than M rises (By a power of 2). This means, that there will still me much more stars on the lower end of the age-scale, than on the higher end.
The three informations I have about any given star, are its mass, its age and its distance from earth. From these three informations I need to derive its fractional main sequence age (τ) and its apparent magnitude (V) and ultimately the probability density for all stars.
Hurley Pols and Tout published a paper in 2000 in which they approximate the stellar evolution as a function of initial mass (M_{ini}), fractional main sequence age (τ) and metalicity(Z). Now I need to know τ. Using equation 5**citation needed** I know the main sequence age (t_{ms}) and τ then becomes: $\tau = \frac{t}{t_{ms}}$. Since I only include stars on the main sequence, I can safely include the condition: $\tau < 1$ to cut down on computing time. Equation 12 and 13**citation needed** are very powerful equations to compute luminosity and radius for a star on the main sequence.

$$\log \frac{L_{MS}(t)}{L_{ZAMS}} = \alpha_L \tau + \beta_L \tau^{\eta} + \left(\log \frac{L_{TMS}}{L_{ZAMS}} - \alpha_L - \beta_L \right) \tau^2 - \Delta L(\tau_1^2 - \tau_2^2) \quad (1)$$

$$\log \frac{R_{MS}(t)}{R_{ZAMS}} = \alpha_R \tau + \beta_R \tau^{10} + \gamma \tau^{40} + \left(\log \frac{R_{TMS}}{R_{ZAMS}} - \alpha_R - \beta_R - \gamma \right) \tau^3 - \Delta R(\tau_1^3 - \tau_2^3) \quad (2)$$

detailed description of these values like α_L?
insert HRD illustrating lumiradius

The program does allow for freely changeable metalicities. For my purposes I use Z=0.02 for all stars to simulate a metalicity similar to that of our galactic neighborhood.
I run this program for every possible mass-age tuple and save the results in a matrix This way I don't have to call the program for every distance-mass-age triple.
With this I now know distance, mass, age, fractional main sequence age, luminosity and radius for any given star. I can now use these informations to compute the apparent magnitudes.

$$M_V = V - 5 \cdot \log_{10}(\mathit{distance}) + 5 - \mathit{Red} \tag{3}$$

$$M_{bol} = M_V + BC \tag{4}$$

$$\frac{L}{L_\odot} = 0.4 \cdot (4.72 - M_{bol}) \tag{5}$$

Where M_V is the absolute visual magnitude, Red is the reddening as a function of distance, M_{bol} is the absolute bolometric magnitude and BC is the Bolometric Correction. Using equations 3, 4 and 5 I can now compute the apparent visual magnitude V:

$$V = 5 \cdot \log_{10}(\mathit{distance}) - 5 + \mathit{Red} + 4.72 - \frac{L}{L_\odot \cdot 0.4} - BC \tag{6}$$

To get an approximation for reddening, I use figure 9 a graphic by AmoresLepine2005, I pick three points in the graph: 1:0.9, 2:2.25 and 3:3.273
I need sources for these formulas.
The next thing I need to know are the probability densities for stars in distance $\left(\frac{dp}{dr}\right)$, mass $\left(\frac{dp}{dm}\right)$ and age $\left(\frac{dp}{dt}\right)$. In my simulation I assume a homogenous distribution of stars. This makes finding a probability density for distance very easy. **Since volume is a function of distance, I can also use** $\left(\frac{dp}{dV}\right)$:

$$\frac{dp}{dV} = \frac{1}{V_{tot}} = \frac{1}{\frac{4}{3} \cdot \pi \cdot \mathit{maxdistance}^3} \tag{7}$$

I assume a constant star formation rate, so the probability density in age would be similar to the density in distance. I do however use a logarithmic binning in age. Thus I can not simply use $\frac{dp}{dt}$ but instead need to find $\frac{dp}{d\log t}$ using the following: $\frac{dp}{dt} = \frac{1}{t_{ms}} \quad \wedge \quad d\log t = \frac{dt}{t \ln 10}$

$$\frac{dp}{d\log t} = \frac{t \ln 10}{t_{ms}} \tag{8}$$

I assume, that the stars are distributed in mass following the salpeter initial mass function: $\frac{dp}{dM} = A \cdot M^{-2.35}$. Similar to the density function in age, I need to convert it for my logarithmic binning.

$$\frac{dp}{d\log M} = \ln 10 \cdot A \cdot M^{-1.35} \tag{9}$$

Where A is a normalization factor, that needs to be computed.

$$1 = \int_{M_{min}}^{M_{max}} A \cdot M^{-2.35} dM = \left[-1.35 \cdot A \cdot M^{-1.35}\right]_{M_{min}}^{M_{max}} \tag{10}$$

$$A = \frac{1.35}{M_{min}^{-1.35} - M_{max}^{-1.35}} \tag{11}$$

With this information I can now formulate an overarching probability density function with regard to τ

$$\frac{dp}{d\tau} = \frac{dp}{dV} dV \cdot \frac{dp}{d\log t} d\log t \cdot \frac{dp}{d\log M} d\log M \cdot \frac{1}{d\tau} \tag{12}$$

2.1 optimization

The binning can be freely adjusted. To optimize runtime of my program I conducted a few tests to adjust the binnings in all three dimensions.
graphics of 50-100-100, 100-50-100, 100-100-50 with caption: this is why I chose 1:2:2

I execute a pair of nested loops in mass and age to get all the data I need from lumiradius.py and save the results in two dimensional arrays. This way I won't have to call the function unnecessarily often in the core function.

3 results and conclusion

–

4 ending