calcPost

Nathan Stein

The logic behind these checks is that the predicted luminosity in filter f for a binary star is

$$\ell_f = 10^{-\texttt{mag[0][f]}/2.5} + 10^{-\texttt{mag[1][f]}/2.5},$$

where `mag[0][f]` and `mag[1][f]` are the magnitudes of the primary and secondary masses, respectively. The predicted magnitude is then calculated as

$$\mu_f = -2.5 \log_{10}(\ell_f).$$

To try to reduce the number of calculations, we can assume that any combination of primary and secondary masses that leads to a predicted magnitude more than $4\sigma_f$ from the observed magnitude will contribute 0 to the posterior. That is, if x_f is the observed magnitude in filter f, we need

$$\mu_f \in [x_f - 4\sigma_f, x_f + 4\sigma_f],$$

or, equivalently,

$$\ell_f \in \left[10^{-(x_f+4\sigma_f)/2.5}, 10^{-(x_f-4\sigma_f)/2.5}\right] \tag{1}$$

if this particular combination of primary and secondary masses will contribute anything to the posterior.

`calcPost` is passed a fixed primary mass, so the purpose of all these checks is to see, for the given primary mass, which choices of secondary mass will lead to a binary with a non-negligible contribution to the posterior, and only compute the posterior for those legitimate secondary mass values.

The first check, `isOverlap`, is designed to test whether *any* secondary mass can lead to an ℓ_f that satisfies (1). Since

$$10^{-\texttt{mag[1][f]}/2.5} > 0,$$

if

$$10^{-(x_f-4\sigma_f)/2.5} - 10^{-\texttt{mag[0][f]}/2.5} < 0,$$

then no ℓ_f can satisfy (1), so there's no need to do any calculations for binary companions. (I think this means there is a bug in the code—we should not set `isOverlap` to 0 if `diffUp` ≤ 0. Also, I think the later code should just skip the log posterior calculations if `isOverlap` is 0. Or `okMass[i]` should be 0 for all i if `isOverlap` is 0.)

Solving (1) for `mag[1][f]`, we have

$$\begin{aligned}
10^{-\texttt{mag[1][f]}/2.5} &\in \left[10^{-(x_f+4\sigma_f)/2.5} - 10^{-\texttt{mag[0][f]}/2.5}, 10^{-(x_f-4\sigma_f)/2.5} - 10^{-\texttt{mag[0][f]}/2.5}\right] \\
10^{-\texttt{mag[1][f]}/2.5} &\in [\texttt{diffUp}, \texttt{diffLow}] \\
\texttt{mag}[1][\texttt{f}] &\in [-2.5\log_{10}(\texttt{diffLow}), -2.5\log_{10}(\texttt{diffUp})] \\
\texttt{mag}[1][\texttt{f}] &\in [\texttt{magLower}, \texttt{magUpper}] \qquad (2)
\end{aligned}$$

The goal of the `okMass` code is to traverse the isochrone to find which secondary masses satisfy (2) in all filters (hence the multiplication by 1—if a particular mass does not satisfy (2) in *any* filter, we ignore it in the posterior calculations). Then, we only calculate a posterior density for the 'ok' secondary masses, truncating the posterior to 0 for any other secondary masses.