sampleMass

Nathan Stein

November 13, 2013

To adapt the algorithm of sampleWDMass to handle main sequence stars and red giants, we need to be able to sample over a 2-D grid (primary mass and mass ratio). Otherwise the algorithm is the same. We have to do the following for each star and each element of sampledPars (the elements of sampledPars are also sent to logPost1Star, as in sampleWDMass.cpp).

```
logPosts ← array[length(primaryMasses), length(massRatios)]
for primaryMass in primaryMasses do
    for massRatio in massRatios do
        star.primaryMass ← primaryMass
        star.massRatio ← massRatio
        logPosts[primaryMassIndex, massRatioIndex] ← logPost1Star(star, ...)
    end for
end for
maxLogPost ← maximum(logPosts)
denom ← sum(exp(logPosts - maxLogPost))
$U$ ← uniform random number in (0,1)
cumulative ← 0
primaryMassIndex ← 0
massRatioIndex ← 0
while cumulative $\leq U$ do
    cumulative += exp(logPosts[primaryMassIndex, massRatioIndex] - maxLogPost) / denom
    massRatioIndex += 1
    if massRatioIndex > length(massRatios) then
        massRatioIndex ← 0
        primaryMassIndex += 1
    end if
end while
sampledPrimaryMass ← primaryMasses[primaryMassIndex]
sampledMassRatio ← massRatios[massRatioIndex]
```

We should also leave the calculation for clusMemPost in there. (Sampling a mass is done conditional on the star being a cluster member; we also want the probability that a star is a cluster member, but that code shouldn't change much from sampleWDMass to sampleMass.)