

TP10 FABRICATION D'UN DIAGRAMME HR

As usual, l'invitation GitHub pour votre dossier de travail vous a été envoyée par mail.

Partie I

Introduction

De longs et complexes programmes ont été écrits pour modéliser l'évolution d'une étoile tout au long de sa vie. Ceux-ci nous donnent de précieuses informations sur les différents stades d'évolution d'une étoile et combien de temps elle passe dans chacun de ces stades. Même s'ils sont très intéressés par ces informations, les astrophysiciens qui s'intéressent à la dynamique stellaire (les mouvements d'étoiles dans des ensembles autogravitants) ont leurs propres programmes pour observer l'évolution de la *position* d'une étoile (alors considérée comme un point matériel) au cours du temps et n'ont pas de ressources CPU à « gâcher » pour faire tourner un programme complet d'évolution stellaire pour chacune des millions d'étoiles qui gravitent dans leurs simulations...

En d'autres termes, les dynamiciens stellaires aimeraient connaître des informations telle que la luminosité de chacune des étoiles de leur simulation à chaque instant sans se fatiguer à évoluer indépendamment chaque étoile à l'aide d'un code d'évolution stellaire dédié. C'est là qu'interviennent les fonctions d'interpolations mises au point par Jarod Hurley et al. ¹ qui permettent, connaissant la masse et l'âge de l'étoile, de donner directement, entre autres choses, la luminosité, la température de surface et la phase d'évolution (voir figure suivante) de l'étoile concernée.

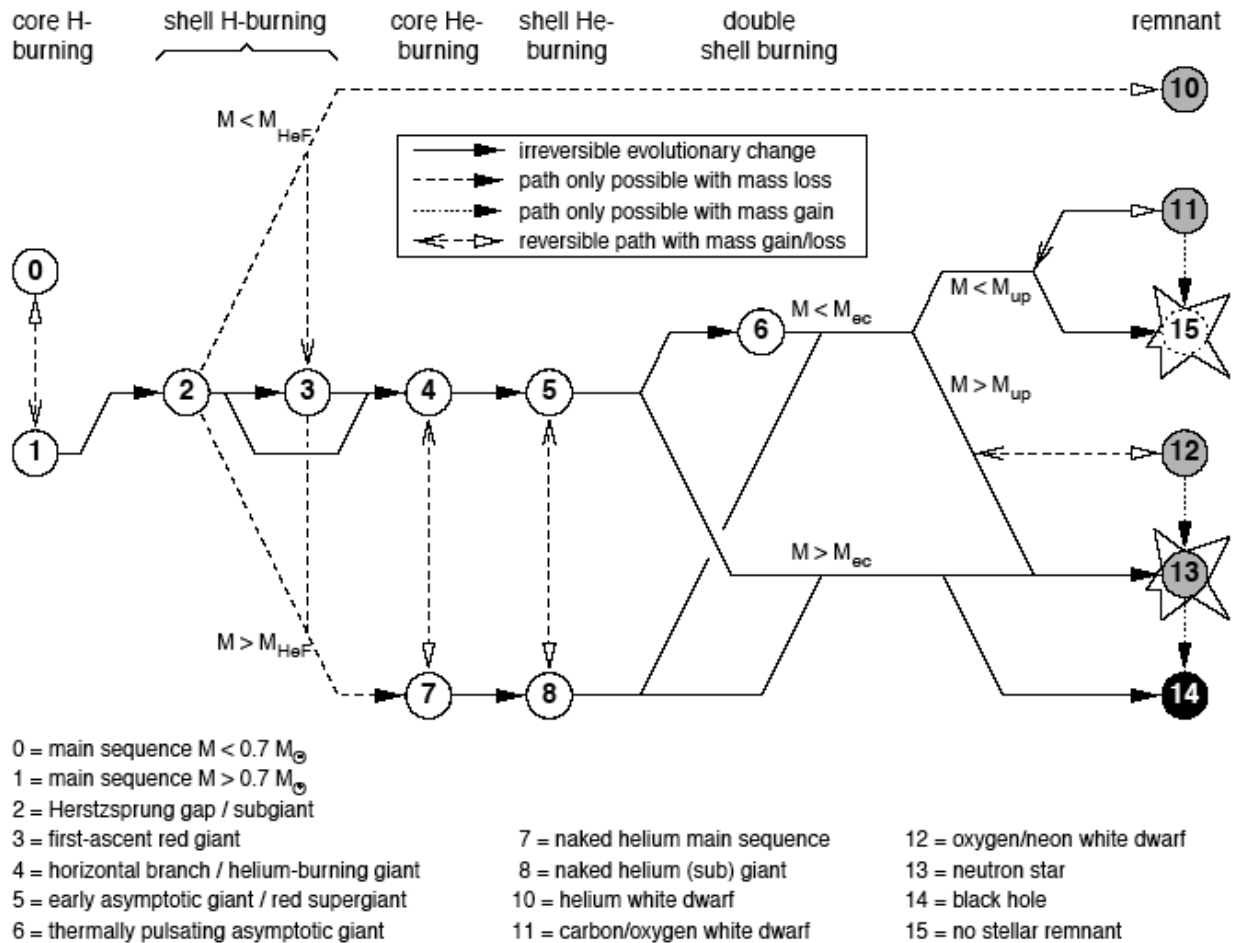


Figure 19. Possible evolution paths through the various stellar evolution phases.

1. Hurley, J. R., Pols, O. R., & Tout, C. A. 2000, MNRAS, 315, 543

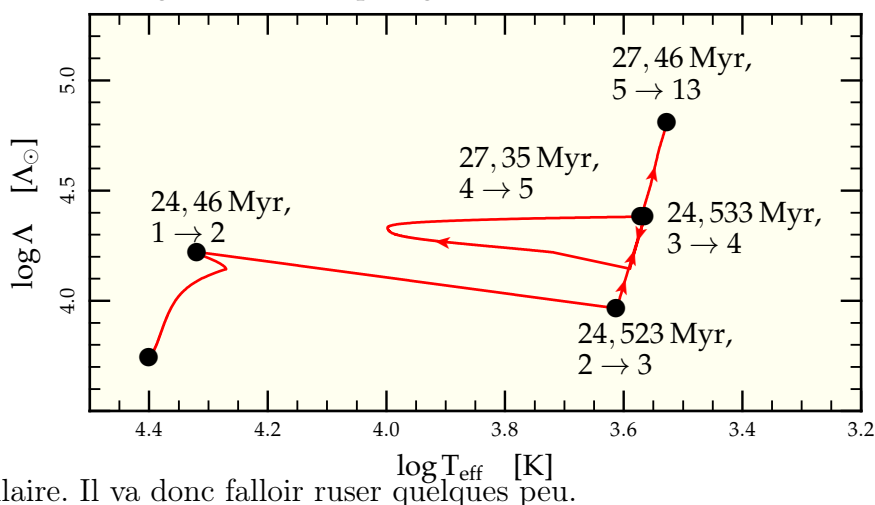
Partie II

Principe du diagramme HR et difficultés

Vous venez d'arriver dans le métier et vous êtes encore méfiants au sujet de ces formules d'interpolation. Vous aimeriez bien pouvoir comparer ces formules à un bon vieux code d'évolution stellaire². Pour cela, le mieux est de représenter l'évolution de l'étoile au cours du temps dans un diagramme dit de Hertzsprung-Russell qui représente le logarithme décimal de la luminosité de l'étoile en fonction³ de l'opposé du logarithme décimal de la température. On pourra alors comparer avec les résultats du code d'Eggleton⁴. La difficulté principale vient du fait que les temps caractéristiques que l'étoile passe dans chacune des phases de sa vie sont très différents en ordre de grandeur. Imaginez que vous suiviez un escargot qui se déplace d'Orléans à Paris à vitesse « normale⁵ ». Naturellement, si vous voulez suivre son périple, vous prendrez une photo satellite toutes les mois environs⁶. Mais si, arrivé à Orly, il embarque soudainement sur un avion pour Strasbourg, il sera arrivé en moins de deux heures et il est peu probable que votre satellite puisse prendre ne serait-ce qu'une photo de son transit...

Pour les étoiles, c'est exactement la même chose. Elles passent l'essentiel de leur temps sur la « séquence principale », c'est-à-dire dans l'état « 1 » de l'évolution stellaire alors que les étapes suivantes se font bien plus vite. Comme ce serait du gachis de prendre une photo toutes les heures durant la première partie du trajet de l'escargot, on ne va pas demander au programme d'évaluer, année après année⁷, si l'étoile a ou non quitté la séquence principale pour s'aventurer dans les autres stades de l'évolution stellaire. Il va donc falloir ruser quelques peu.

Diagramme Hertzsprung-Russell d'une étoile $m = 10 M_{\odot}$



Partie III

Angle d'attaque

Pour explorer ces routines, vous disposez du module `sse` stocké sous forme compilé dans le répertoire TP10/ et plus particulièrement de la fonction `sse.get_evolution(t)` qui donne les informations sur l'état de l'étoile à la date `t`, sa luminosité et sa température de surface. Par exemple, vous pouvez utiliser

```
1 import sse # Import du module
2 help(sse.get_evolution) # Obtenir l'aide correspondant à la fonction appelée
3 t = 10 # On regarde à t = 10 Myr
4 s,L,T = sse.get_evolution(t) # Qu'obtient-on ?
5 print("À t={t} Myr, état {s} avec log10(L)={L} et log10(T)={T}".format(t,s,L,T))
```

2. Le plus renommé, celui de Peter Eggleton, date des débuts des années 70.

3. Pour des raisons historiques...

4. Revisité depuis les années 70 et que l'on peut accéder sur le web via l'interface EZweb :

<http://www.astro.wisc.edu/~townsend/static.php?ref=EZ-Web>

5. Pour un escargot bien sûr...

6. À raison de 1 mm/s de vitesse moyenne, il lui faudra environ 1 284 jours = 3.5 années pour parcourir les 111 km qui séparent les deux villes en ligne droite.

7. L'échelle de temps « de base » en évolution stellaire est au moins de l'ordre du million d'années (pour les étoiles de forte masse $> 2 M_{\odot}$), voire du milliard d'années (pour les étoiles de faible masse $\lesssim 2 M_{\odot}$).

Help on function `get_evolution` in module `sse`:

`get_evolution(t)`

Fonction qui donne l'état d'une étoile de 7 masses solaires à l'instant `t` demandé (exprimé en Myr = million d'années). Elle renvoie un triplet `(s,L,T)` où

- * `s` est un entier représentant le status de l'étoile (voir figure du TP) ;
- * `L` est le logarithme décimal de la luminosité de l'étoile (exprimée en luminosités solaires) à cet instant ;
- * `T` est le logarithme décimal de la température de surface de l'étoile (exprimée en kelvins) à cet instant.

Si vous demandez un temps "hors limite", la fonction renvoie `(None,None,None)`

À `t=10` Myr, état 1 avec `log10(L)=3.279650664594495` et `log10(T)=4.308693055498753`

La durée de vie totale de l'étoile à $7 M_{\odot}$ que vous désirez étudier est d'environ 60 Myr. Le but du TP est, dans un premier temps, de dessiner le diagramme HR de l'étoile en plaçant 50 points régulièrement espacés en temps sur chacune des phases (donc 300 points au final vu qu'il y aura 6 phases, les formules d'interpolations n'allant, dans notre cas, que jusqu'à l'état de géante rouge pulsante sur la branche asymptotique des géantes). Pour ce faire, voici la marche à suivre :

1. Écrire une fonction `dates_transitions()` qui renvoie une liste à 6 valeurs où chaque valeur représente l'âge de transition de l'étape i à l'étape $i + 1$ (la dernière valeur étant l'âge limite pour lequel les formules d'interpolation renvoient une valeur sensée [et non `None`]). On pourra s'inspirer des techniques développées en cours concernant la recherche par dichotomie d'une valeur dans une liste ou du zéro d'une fonction.

STOP GitHub

Allez sur Github Desktop pour faire un commit. Choisissez vous-même (avec pertinence) le résumé. Pensez aussi à appuyer sur le bouton «Push origin» en haut à droite pour mettre à jour sur le web.

2. Écrire une fonction `obtiens_valeurs_T_L(transitions)` qui prend en entrée la liste à 6 valeurs précédemment calculée et renvoie deux tableaux de 300 éléments chacun contenant les valeurs en $\log T$ et $\log L$ correctement échantillonnées à raison de 50 points sur chaque phase⁸, c'est-à-dire que les points de 0 à 49 appartiennent à la phase 1, les points de 50 à 99 à la phase 2, etc. Pour les tests, la liste de transitions donnée est telle que si $t_{i,i+1}$ correspond à la date de transition de l'état i à l'état $i + 1$, alors l'état de l'étoile en $t_{i,i+1}$ est l'état $i + 1$.

STOP GitHub

Allez sur Github Desktop pour faire un commit. Choisissez vous-même (avec pertinence) le résumé. Pensez aussi à appuyer sur le bouton «Push origin» en haut à droite pour mettre à jour sur le web.

3. Enfin, écrire une fonction `diagramme_HR(T,L)` qui prend en entrée les deux listes calculées précédemment et qui dessine en sortie le diagramme HR proprement dit à sauvegarder dans le fichier `HR_sse_VotreNom.png` (mettez bien sûr votre propre nom sans accent ni espace).

STOP GitHub

Allez sur Github Desktop pour faire un commit. Choisissez vous-même (avec pertinence) le résumé. Pensez aussi à appuyer sur le bouton «Push origin» en haut à droite pour mettre à jour sur le web.

8. NB : 50 est la valeur par défaut du nombre de points donnés par `np.linspace()`.

Partie IV

Comparaison avec EZ-web

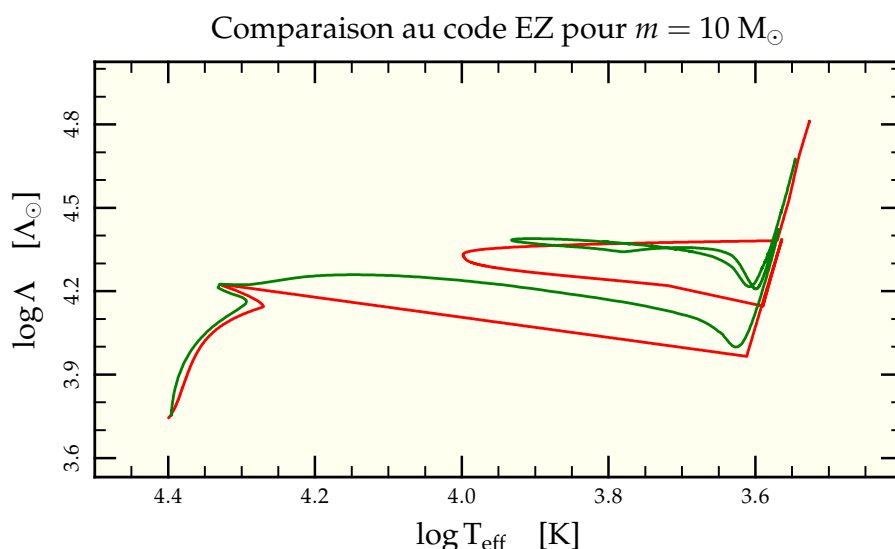
Dessiner un diagramme HR n'est pas tout. Encore faut-il comparer le résultat avec un code qui aura pris un soin tout particulier à inclure tout ce que l'on sait sur les mécanismes physiques à l'origine de l'évolution stellaire. Vous avez récupéré une telle simulation dans le fichier `TP10/EZ_summary.txt` qui contient énormément d'informations (29 colonnes en tout). Il faut à présent le lire, en extraire ce qui vous intéresse (colonnes 4 pour $\log L$ et 6 pour $\log T$) et superposer le diagramme HR correspondant à votre précédent diagramme. On procèdera donc comme suit :

1. Écrire une fonction `lis_EZ()` qui lit le fichier `TP10/EZ_summary.txt` et renvoie un triplet $(\mathbf{t}, \mathbf{L}, \mathbf{T})$:
 - \mathbf{t} est une liste contenant tous les âges présents dans la simulation (2^e colonne du fichier).
 - \mathbf{L} est une liste contenant les valeurs de $\log L$ en correspondance avec les âges contenus dans \mathbf{t} (4^e colonne du fichier).
 - \mathbf{T} est une liste contenant les valeurs de $\log T$ en correspondance avec les âges contenus dans \mathbf{t} (6^e colonne du fichier).

STOP GitHub

Allez sur Github Desktop pour faire un commit. Choisissez vous-même (avec pertinence) le résumé. Pensez aussi à appuyer sur le bouton «Push origin» en haut à droite pour mettre à jour sur le web.

2. Écrire une fonction `diagramme_HR_total()` qui va superposer sur un même diagramme les résultats des formules d'interpolation et ceux de la simulation EZ. Cela devrait vous donner un résultat comme celui-ci (pour $m = 10 M_{\odot}$ et non $7 M_{\odot}$ comme dans vos données). Sauvegarder le résultat dans le fichier `HR_EZ_VotreNom.png` (mettez bien sûr votre propre nom sans accent ni espace).



STOP GitHub

Allez sur Github Desktop pour faire un commit. Choisissez vous-même (avec pertinence) le résumé. Pensez aussi à appuyer sur le bouton «Push origin» en haut à droite pour mettre à jour sur le web.

On pourra terminer en comparant les âges de transition dans les deux modèles, c'est-à-dire en regardant la position dans le diagramme HR de EZ à l'âge correspondant à chaque transition tiré des formules d'interpolation.