RD-3 RTIGE-OPM-YB-03.08 Étoiles du champ

Yannick Boissel

8 décembre 2011

Table des matières

I Introduction **2**
0.1 Ajouts depuis la version 7 . . . 3
0.2 Introduction . . . 3

II Contenu de l'archive **4**
0.3 Fichiers de contenu . . . 5
0.3.1 Dossier Modele_galaxie . . . 5
0.3.2 Dossier HR . . . 6

III Méthode **8**
0.4 Etoiles indiscernables . . . 9
0.4.1 Méthode . . . 9
0.4.2 Petits plus . . . 9
0.5 Etoiles discernables . . . 11
0.5.1 En utilisant l'âge moyen des particules . . . 11
0.5.2 En donnant un âge différent à chaque étoile . . . 11
0.6 Montrer l'évolution des étoiles . . . 12
0.7 Naines brunes . . . 12

IV Journal de bord **13**
0.8 Introduction . . . 14
0.9 Journal de Bord . . . 15

Première partie

Introduction

0.1 Ajouts depuis la version 7

Quelques erreurs corrigées dans la description du format des fichiers de données.

0.2 Introduction

Ce document explique comment simuler les étoiles qui ne sont pas dans des amas. Le format des données a été choisi pour maximiser la précision des informations scientifiques tout en minimisant la mémoire utilisée et le temps de calcul nécessaire. Egalement, il tient compte de l'avenir du projet, puisqu'il permet de faire vieillir ou rajeunir les étoiles. Ce format de données permet à la fois de voir la Galaxie de loin, sans discerner les étoiles, ou bien de près en discernant les étoiles, et ces deux approches sont indépendantes.
En utilisant des histogrammes de diagrammes HR calculés à des âges différents et des cartes de paramètres stellaires, calculés à partir d'isochrones, on réduit d'un facteur 40 la mémoire utilisée. Les fichiers d'isochrones pèsent dans les 300 Mo s'ils incluent les étoiles mortes, et les cubes de données que j'ai construits pèsent dans les 7 Mo, et ils incluent les étoiles mortes.
Ce format est également utilisé pour les amas d'étoiles, afin d'uniformiser et de simplifier les formats de données scientifiques.

Les données permettent de simuler le disque (vieux), le bulbe (vieux) et le halo. Les étoiles jeunes, la formation stellaire et les gaz et poussières seront traitées dans le document "Milieu interstellaire".

Dans l'archive, outre ce document, il y a deux dossiers :

- 'Modele_galaxie' : il contient les fichiers du modèle de galaxie (étoiles seulement).
- 'HR' : il contient de quoi simuler les paramètres stellaires pour des particules ou des étoiles (particules éclatées).

Tous les fichiers sont en ASCII. Tous les cubes sont arrangés ainsi : pour chaque tranche, l'origine est en haut à gauche, et les abscisses sont écrites verticalement de haut en bas, les ordonnées horizontalement de gauche à droite (les tableaux sont tournés de 90°). Ce que j'appelle abscisse est l'abscisse du tableau dans le code. Il se trouve que le tableau est tourné au moment de l'écriture.

Il se présente deux cas de figure pour la simulation des étoiles de champ : Le cas où l'on ne discerne pas d'étoiles, le cas où l'on discerne des étoiles. Cette version du contenu permet de simuler entièrement le cas 1 et partiellement le cas 2. Lire la méthode donnée pour le cas 1) avant celle donnée pour le cas 2).

Deuxième partie

Contenu de l'archive

0.3 Fichiers de contenu

0.3.1 Dossier Modele_galaxie

Ce dossier contient 22 fichiers de modèle de galaxie et un fichier contenant une carte d'âges d'étoiles. Il y en a de deux types : les "stars_disc", qui concernent les particules du disque, et les "stars_bulb" qui concernent les particules du bulbe et du halo. Il y a 11 fichiers de chaque type correspondant à 11 âges de la Galaxie, de 12.5 à 13 milliards d'années, par pas de 50 millions d'années. Le fichier 00 a un âge de 12.5 Gyr.

– **'RTIGE_2011_CombesDiMatteo_stars_disc_sft##.dat'** ou
'RTIGE_2011_CombesDiMatteo_stars_bulb_sft##.dat' : Dans ces fichiers on trouve un ensemble de particules, toutes de masse totale identique (Elle est donnée dans les fichiers), avec comme données :

0. position x en kpc (par rapport au centre galactique)
1. position y en kpc (par rapport au centre galactique)
2. position z en kpc (par rapport au centre galactique)
3. composante vx de la vitesse en 100 km/s
4. composante vy de la vitesse en 100 km/s
5. composante vz de la vitesse en 100 km/s
6. masse de la particule en $2.3 \times 10^9\ M_\odot$
7. nombre total d'étoiles que l'on trouve en moyenne pour une particule de cette masse.
8. métallicité moyenne Z

9. date moyenne de naissance des étoiles de la particule en 10^7 yr
10. âge de la particule en $log10(yr)$

Pour les fichiers sft00, pour le disque, les âges tournent autour de valeurs espacées de 0.01 et allant de 6.60 à 9.98. Pour le halo, l'âge est de 12.5 Gyr (10.10 dans les unités du fichier) et pour le bulbe il est de 11.5 Gyr (10.06 dans les unités du fichier). Les fichiers suivants ont des âges plus vieux : 50 millions d'années supplémentaires sont ajoutées aux particules de chaque nouveau fichier plus âgé.

Ages :
Pour représenter la variété d'âges des étoiles dans le disque, j'ai donné aux particules du modèle 00 un âge selon son altitude au plan du disque, à partir de la carte d'âge fournie dans cette archive.

Métallicités :
J'ai discrétisé les valeurs de métallicité qui étaient données par le modèle afin de l'alléger. J'en ai choisi 6 : 0.005, 0.01, 0.015, 0.02, 0.025, 0.03. Pour chaque valeur, j'ai regardé quelles sont les particules qui ont une métallicité à ± 0.0025, et je leur ai donné cette valeur. Cependant, la première tranche va de 0 à 0.0075, et la dernière de 0.0275 à 0.04. Pour le halo et le bulbe, j'ai pris trois valeurs : $Z = 0.0004$, $Z = 0.0008$ et $Z = 0.0002$ que j'ai réparties aléatoirement en tirant pour chaque particule l'une de ces trois valeurs selon une loi uniforme. J'ai donc en tout 6+3=9 valeurs de métallicités. Voir Journal de bord au 15 novembre 2011.

– **'RTIGE_2011_agemap.dat' :** Dans ce dossier on trouve également le fichier 'RTIGE_2011_agemap.dat', qui est une carte des âges des étoiles du disque en fonction de l'altitude. Dans chaque case est donné le nombre relatif d'étoiles. Les valeurs de cet histogramme sont normalisées à 1. L'origine est en haut à gauche, et les abscisses sont écrites verticalement de haut en bas, les ordonnées horizontalement de gauche à droite (le tableau est tourné de 90°) En abscisses, on a l'âge dans l'unité log10(années), de 6.60 à 10.00, par pas de 0.01, et en ordonnées on a l'altitude, de 0 à

4000 pc, par pas de 50 pc.
La distribution d'âges en fonction de l'altitude utilisée pour construire la carte d'âges est :

$$f(z,a) = \alpha e^{-z/100a} \qquad a \leqslant 3\ Gyr \tag{1}$$
$$\alpha e^{-z/300} \qquad a > 3\ Gyr \tag{2}$$

(pour $z \geqslant 0$) a en Gyr, z en pc. Avec α un facteur de normalisation (l'intégrale de f doit faire 1) :

$$\alpha = \frac{1}{300 - 50a_1^2 + 300a_2 - 900} \tag{3}$$

L'altitude z est intégrée entre 0 et $+\infty$ et l'âge entre $a_1 = log10(6.60)/10^9$ et $a_2 = log10(10.00)/10^9$ pour le calcul de α. La carte se limite à $z = 4000$ pc : c'est acceptable car au-delà de 4000 pc, le nombre d'étoile tombe à quelques unes. J'ai mis un âge de 10.00 (en log10(années)) aux étoiles au-dessus de 4000 pc dans le modèle de Fux augmenté que je fournis. Voir journal de bord au 21 septembre 2011 pour plus de détails.

0.3.2 Dossier HR

Dans le dossier 'HR', on trouve pour 9 métallicités (z dans le nom des fichiers), 4 fichiers différents :

– **'RTIGE_2011_HRcube_z#.dat' :** Ce fichier est constitué de 354 tableaux 100x100 au format entier 16 bits. Les tableaux sont écrits à la file, séparés par un #. Chaque tableau est un diagramme H-R calculé pour un âge donné : il ne contient qu'une seule isochrone, la formation stellaire étant terminée dans les particules étoiles. Pour chaque tranche, l'origine est en haut à gauche, et les abscisses sont écrites verticalement de haut en bas, les ordonnées horizontalement de gauche à droite (les tableaux sont tournés de 90°).

  En ordonnées on a le log10 de la température, en $log10(K)$, croissant de bas en haut, de 3.3 à 4.7, et en abscisse, on a le log10 de la luminosité en bande V, en $log10(L_\odot)$, croissant de gauche à droite, de -4.0 à 6.5
  Chaque case donne le log10 de la proportion d'étoiles sur 1 million, fois 1000, ayant ce couple $(log10(L_V), log10(Te))$. Ces valeurs sont donc comprises entre 0 et 6000. Si les diagrammes HR sont donnés avec des nombres relatifs d'étoiles dans une unité si étrange, c'est parce que cela m'a permis de les enregistrer au format entier 16 bits, en perdant peu d'information, produisant ainsi des fichiers les plus légers possible.
  Les 354 tableaux correspondent à 354 âges, espacés de 0.01 dans l'unité $log10(annees)$, la même unité que pour les données du modèle de galaxie.

– **'RTIGE_2011_star_params_z#.dat' :** Ce fichier est constitué de 15 tableaux 100x100 au format float 64 bits. Les tableaux sont écrits à la file, séparés par un #. Il s'agit de cartes de paramètres stellaires, correspondant aux diagrammes HR. Ainsi, pour chaque case de diagramme HR, on dispose de 15 paramètres stellaires en plus de la température et de la luminosité. Pour chaque tranche, l'origine est en haut à gauche, et les abscisses sont écrites verticalement de haut en bas, les ordonnées horizontalement de gauche à droite (les tableaux sont tournés de 90°). Ces paramètres stellaires sont dans l'ordre :

  0. log10(Luminosité bolométrique) en $log10(L_\odot)$
  1. log(gravité) en $log10(cm.s^{-2})$
  2. masse stellaire en $M_\odot$
  3. rayon stellaire en $R_\odot$
  4. magnitude absolue en bande U
  5. magnitude absolue en bande B
  6. magnitude absolue en bande V
  7. magnitude absolue en bande R

8. magnitude absolue en bande I
9. magnitude absolue en bande J
10. magnitude absolue en bande H
11. magnitude absolue en bande K
12. couleur normalisée : rouge
13. couleur normalisée : vert
14. couleur normalisée : bleu

– **'RTIGE_2011_lcmap_z#.dat' :** Ce fichier contient les classes de luminosité des étoiles situées dans les cases de diagramme HR. Il y a 100x100 chaines de caractères, le tableau est orienté dans le même sens que les diagrammes HR et les cartes de paramètres stellaires. Attention, suite à une erreur de code, les lignes sont pour l'instant encadrées de crochets, au lieu d'avoir un tableau écrit sous forme d'une matrice.

– **'RTIGE_2011_spcmap_z#.dat' :**Ce fichier contient les classes spectrales des étoiles situées dans les cases de diagramme HR. Il y a 100x100 chaines de caractères, le tableau est orienté dans le même sens que les diagrammes HR et les cartes de paramètres stellaires. Attention, suite à une erreur de code, les lignes sont pour l'instant encadrées de crochets, au lieu d'avoir un tableau écrit sous forme d'une matrice.

Troisième partie

Méthode

0.4 Etoiles indiscernables

0.4.1 Méthode

1) Choisir l'un des modèles de galaxie : cela revient à choisir l'âge de la Galaxie entre 12.5 et 13 Gyr.

2) Pour chaque particule du disque et du bulbe (dont halo), choisir les fichiers 'RTIGE_2011_HRcube_z#.dat' et 'RTIGE_2011_star_params_z#.dat' correspondant à leur métallicité.

3) Pour chaque particule, lire le diagramme HR correspondant à son âge.

4) Le nombre total d'étoiles pour chaque particule est donnée dans les fichiers du modèle de Galaxie.

5) Maintenant, il reste à calculer pour chaque particule, sa luminosité totale et sa couleur moyenne, afin de la représenter dans la visualisation. Il faut normaliser les diagrammes HR à 1 avant de les utiliser. Ils sont calculés à partir d'un ensemble de 10^6 étoiles, et les proportions d'étoiles sont données en log10, puis multipliées par 1000. On a donc :

$$Iso_{norm} = \frac{10^{Iso/1000}}{10^6} \tag{4}$$

6)Pour avoir la luminosité totale, il suffit de faire la somme des abscisses de toutes les cases i du diagramme HR, pondérée par les valeurs n_i des cases du diagramme HR, fois N, le nombre d'étoiles dans la particule (la normalisation est déjà incluse dans l'équation) :

$$L_{V_{tot}} = \left[\sum_{i}^{nb_c} \frac{10^{n_i/1000}}{10^6} \times 10^{x_i} \right] \times N \tag{5}$$

Avec $x_i = log10(L_{V_i})$ et nb_c le nombre total de cases de l'histogramme de diagramme HR.

7) Afin d'avoir la luminosité de la particule dans les 3 canaux de couleur, il faut récupérer les cartes de couleurs dans le fichier 'RTIGE_2011_star_params_z#.dat'.

$$L_{V_{r_{tot}}} = \left[\sum_{i}^{nb_c^*} \frac{10^{n_i/1000}}{10^6} \times r_i \times 10^{x_i} \right] \times N \tag{6}$$

Avec r_i la valeur de la couleur normalisée r pour la case i, et n_i la valeur de l'histogramme de diagramme HR pour la case i.
* : nb_c est ici le nombre total de cases de la carte de couleur qui correspondent à des cases *non vides* dans l'histogramme de diagramme HR. On fait de même pour les autres couleurs.

8) Calculer les magnitudes apparentes en tenant compte de l'extinction. (Voir le document "Amas globulaires", ou le document "Amas ouverts")

0.4.2 Petits plus

8) Si l'on veut simuler la Galaxie à un âge plus ou moins avancé que ce que fournit le modèle, il suffit d'ajouter un peu d'âge aux particules et de changer de diagrammes HR. Cependant, la formation stellaire antérieure ou ultérieure au modèle ne sera pas prise en compte.

9) Si l'on veut changer de longueur d'onde d'observation (regarder dans une autre bande que V), soit dans une bande donnée X, il faut pour chaque case non vide de l'histogramme de diagramme HR choisi, regarder quelle est la magnitude absolue en bande X correspondante dans le fichier 'RTIGE_2011_star_params_z#.dat' et calculer la luminosité dans la bande X :

$$L_X = 10^{0.4(4.74-M_X)} \tag{7}$$

M_X étant la magnitude absolue de l'étoile ou de la particule en bande X et 4.74 étant la magnitude absolue bolométrique du Soleil. L_X est en $L_\odot$. Il faut ensuite utiliser de fausses couleurs.

Attention, il faut recalculer l'extinction dans cette nouvelle bande. Voir le document "Milieu Interstellaire".

0.5 Etoiles discernables

0.5.1 En utilisant l'âge moyen des particules

Pour l'instant, considérons que l'âge des étoiles dans une particule est identique pour toutes les étoiles, soit l'âge moyen de la particule. On fera cette approximation pour le sphéroïde. Soit un cube de volume V et de côté d, contenant N_P particules. Ce cube peut par exemple être un cube de la grille 3D utilisée dans le logiciel de RSA cosmos.

Le modèle de Fux donne une masse $M_S = 2.48 \times 10^{10} M_\odot$ identique pour chaque particule du sphéroïde.
Soient dN une variable représentant les valeurs de l'histogramme de diagramme HR (correspondant à l'âge moyen de la particule), $\log(L_V/L_\odot)$, les valeurs de luminosité (en log décimal), et $\log(Te)$ les valeurs de température (en log décimal).
On va éclater ces N_P particules une par une dans le cube en respectant les proportions données par les diagrammes HR correspondants, jusqu'à obtenir la masse M_S, si la particule est dans le sphéroïde, ou M_D si elle est dans le disque, selon l'algorithme suivant :

$m = 0$
$i = 0$
while $m < M_S$:
  $x[i]$ =valeur tirée aléatoirement selon une distribution uniforme entre 0 et d.
  $y[i]$ =valeur tirée aléatoirement selon une distribution uniforme entre 0 et d.
  $z[i]$ =valeur tirée aléatoirement selon une distribution uniforme entre 0 et d.
  $j = 0$
  while $j < 1$:
    $\log(l_V/L_\odot)$ =valeur tirée aléatoirement selon une distribution uniforme
      entre $\log(L_V/L_\odot)_{min}$ et $\log(L_V/L_\odot)_{max}$.
    $\log(te)$ =valeur tirée aléatoirement selon une distribution uniforme
      entre $\log(Te_{min})$ et $\log(Te_{max})$.
    dn =valeur tirée aléatoirement selon une distribution uniforme entre 0 et dN_{max}.
    dN_{tmp} =valeur de dN pour le couple $(\log(L_V/L_\odot), \log(Te))$ le plus proche de $(\log(l_V/L_\odot), \log(te))$
      dans l'histogramme.
    m_{tmp} = masse pour le couple $(\log(L_V/L_\odot), \log(Te))$ ainsi repéré, prise dans la carte de masse.
    if $dn <= dN_{tmp}$:
      On retient les valeurs $\log(l_V/L_\odot)$ et $\log(te)$, créant ainsi une nouvelle étoile.
      On retient également les valeurs des autres paramètres donnés par
      le modèle pour le couple $(\log(L_V/L_\odot), \log(Te))$.
      $j+ = 1$
  $i+ = 1$
  $m+ = m_{tmp}$

Les variables indiquées en minuscules sont temporaires, les variables indiquées en majuscule sont liées au diagramme HR utilisé comme référence.

0.5.2 En donnant un âge différent à chaque étoile

L'algorithme sera un peu différent si l'on donne un âge différent à chaque étoile, pour une particule qui est dans le disque. Pour une particule qui est dans le sphéroïde, on prend un âge identique pour chaque étoile, celui de la particule, et on utilise l'algorithme précédent.

Prenons une particule du disque, de masse totale M_D, et d'altitude z au plan du disque. Dans la carte d'âges, que l'on trouve dans le fichier 'RTIGE_2011_agemap.dat', repérer la boîte dans laquelle se situe l'altitude z, et sélectionner la ligne correspondante en âges. Soit $dN_{age_{max}}$ la valeur maximale de cette ligne d'histogramme. Soit $dN_{HR_{max}}$ la valeur maximale de l'histogramme de diagramme HR correspondant à l'âge de l'étoile tirée.

Voici l'algorithme à utiliser :

$m = 0$
$i = 0$
while $m < M_D$:
  $x[i]$ =valeur tirée aléatoirement selon une distribution uniforme entre 0 et d.
  $y[i]$ =valeur tirée aléatoirement selon une distribution uniforme entre 0 et d.
  $z[i]$ =valeur tirée aléatoirement selon une distribution uniforme entre 0 et d.
  $j = 0$
  while $j < 1$:
    a =âge tiré aléatoirement selon une distribution uniforme entre 6.60 et 10.00 (unité : log10(années)).
    dn_{age} =valeur tirée aléatoirement selon une distribution uniforme entre 0 et $dN_{age_{max}}$.
    dN_{age} =valeur de la carte d'âges pour l'âge a.
    if $dn_{age} <= dN_{age}$:
      On retient l'âge a.
      Lire l'histogramme de diagramme HR correspondant à l'âge a.
      $k = 0$
      while $k < 1$:
        $\log(l_V/L_\odot)$ =valeur tirée aléatoirement selon une distribution uniforme entre $\log(L_V/L_\odot)_{min}$ et $\log(L_V/L_\odot)_{max}$.
        $\log(te)$ =valeur tirée aléatoirement selon une distribution uniforme entre $\log(Te_{min})$ et $\log(Te_{max})$.
        dn_{HR} =valeur tirée aléatoirement selon une distribution uniforme entre 0 et $dN_{HR_{max}}$
        dN_{HR} =valeur de l'histogramme de diagramme HR pour le couple $(\log(L_V/L_\odot), \log(Te))$ le plus proche de $(\log(l_V/L_\odot), \log(te))$ dans l'histogramme de diagramme HR
        m_{tmp} = masse pour le couple $(\log(L_V/L_\odot), \log(Te))$ ainsi repéré, prise dans la carte de masse.
        if $dn_{HR} <= dN_{HR}$:
          On retient les valeurs $\log(l_V/L_\odot)$ et $\log(te)$, créant ainsi une nouvelle étoile.
          On retient également les valeurs des autres paramètres donnés par les cartes de paramètres.
          $k+ = 1$
      $j+ = 1$
  $i+ = 1$
  $m+ = m_{tmp}$

0.6 Montrer l'évolution des étoiles

Avec le contenu fourni, il est possible de simuler la galaxie ou un ensemble d'étoiles avec un certain âge. On peut aussi faire évoluer des particules dans le temps en sautant d'un diagramme HR à l'autre. On peut également faire vieillir ou rajeunir des étoiles individuelles. Voici la méthode pour faire passer une étoile d'un âge a_1 à un âge a_2 :

1) Simuler un ensemble d'étoiles avec l'un des algorithmes donnés dans la section précédente.
2) Soit une étoile ayant pour âge a_1. On connaît sa masse grâce à la carte de masse. Lire l'histogramme de diagramme HR correspondant à l'âge a_2.
3) Trouver les cases non-vides dans cet histogramme. Lire les cases correspondantes dans la carte de masse.
4) Chercher dans les masses sélectionnées la masse la plus proche de celle de l'étoile choisie. Retenir les $log10(L_V)$ et $log10(Te)$ correspondant à cette masse dans le diagramme HR associé à l'âge a_2, ainsi que les autres paramètres stellaires correspondant à cette case dans les autres cartes si nécessaire.

0.7 Naines brunes

Les naines brunes sont déjà présentes dans les données actuelles : elles sont contenues dans le groupe d'étoiles de moins de 0.15 $M_\odot$ (hors naines blanches). Cependant, on leur a pour l'instant donné les paramètres stellaires d'une étoile de 0.15 $M_\odot$. Ce que l'on a à faire, c'est répartir toutes les étoiles de moins de 0.15 $M_\odot$ dans diverses sous boîtes. D'après Chabrier 2003, la densité de naines brunes dans le disque est de $n_{BD} \sim 0.1\ pc^{-3}$.

Quatrième partie

Journal de bord

0.8 Introduction

Pour représenter les étoiles qui ne sont pas dans des amas, il faut utiliser des modèles de la Galaxie. Pour les positions des étoiles des différents ensembles de la Galaxie, on utilise le modèle de Fux. Pour récupérer les paramètres stellaires je vais utiliser le modèle de Besançon.

Voilà le mail que j'ai envoyé à Mikaël Lemercier et qui résume ce que je prévois de faire :

"Salut Mikaël,

Pour la luminosité : On ne peut pas se baser uniquement sur la masse des étoiles, cela ne marcherait que pour la séquence principale, et donc pas pour les naines blanches et les géantes et super-géantes rouges.

Le but est d'avoir la luminosité et la couleur d'un paquet d'étoiles d'une masse et d'une quantité d'étoiles données. Le modèle de galaxie de Françoise Combes donne les masses de paquets d'étoiles. Le modèle de Fux ne les donne pas. Si N est le nombre total de paquets présents dans le modèle, et M le nombre d'étoiles de la galaxie, M/N est le nombre moyen d'étoiles dans un paquet.

Voilà ce qu'on peut faire : Simuler une distribution de paramètres stellaires de M/N étoiles (Luminosité, masse, couleurs, taille) pour un paquet donné connaissant la distribution "standard" des étoiles du champ (en dehors des amas) de la Voie Lactée. Tu n'aurais ensuite qu'à effectivement sommer les luminosité de chacune de ces M/N étoiles pour avoir la luminosité d'un paquet, et moyenner leurs couleurs pour avoir la couleur du paquet. Le principe est le même pour la galaxie vue à des distances intermédiaires. Il suffit de ne prendre qu'une partie de l'échantillon d'étoiles. En pratique (si on utilise Fux) :
1) Il faut simuler une distribution d'étoile pour des paquets situés dans 4 grands ensembles : le disque mince, le disque épais, le bulbe, et le halo. Ce qui nous fait 4 "modèles" de paquets.
2) Pour plus de réalisme, il faudrait que dans un grand ensemble donné, deux paquets aient de légères différences dans les étoiles qui les constituent. Pour cela il y a deux solutions : soit je te donne un modèle de paquet pour chaque grand-ensemble et les fluctuations à ajouter d'un paquet à l'autre ; soit je te donne pour chaque grand ensemble, 4 ou 5 modèles de paquets que tu choisis aléatoirement d'un paquet à l'autre. Il faut voir ce qui est le plus pratique et le plus rapide pour le temps-réel.
3) Quand on éclate un paquet en M/N étoiles individuelles, il faut tirer aléatoirement et uniformément les positions de ces étoiles dans une sphère d'une taille donnée (je vais calculer cette taille). Il faut faire un tirage différent pour chaque paquet pour plus de réalisme, sinon ça ferait un effet "papier peint" ;-)
4) Pour la galaxie vue depuis des distances intermédiaires, il faut se placer dans un paquet donné, faire le tirage des étoiles et n'en garder qu'une partie selon le champ de vision accessible, puis faire l'image en sommant les luminosités.

Donc en fait, le système serait le même que pour les amas globulaires ou les amas ouverts (qui seraient en fait les 5ème et 6ème grands-ensembles d'étoiles de la galaxie) : Tu aurais un catalogue de positions de paquets d'étoiles, le modèle de Fux, et divers modèles à répartir pour ces paquets.

Voilà, dis-moi si mon raisonnement est clair.

Pour l'extinction, je vais en discuter avec Françoise Combes (LERMA) et Rosine Lallement (GEPI). Il faut trouver la bonne loi d'extinction. En fait, ce n'est pas le gaz qui cause le plus d'extinction (un peu), mais la poussière. Il est possible que dans le modèle de Françoise Combes et dans Fux, la partie "gaz" représente à la fois le gaz et la poussière. Je vais aussi vérifier ça.

Voilà, je continue de réfléchir à ces questions de mon côté. Pour te fournir les modèles de paquets d'étoiles, je vais utiliser le modèle de galaxie de Besançon."

0.9 Journal de Bord

29 avril 2011

Dans le modèle de Fux, je vois 4 fichiers :
- disc.dat, qui regroupe le disque mince, le disque épais, et une partie du centre galactique.
- sphero.dat, qui donne les étoiles du sphéroïde (bulbe ?)
- darkhalo.dat, qui donne le halo de matière noire
- gas.dat qui donne le gaz et la poussière.
- Où sont les étoiles du halo ? => sphéroïde = bulbe + halo Carine Babusiaux m'a dit.

Nouveau mail de Mikaël Lemercier, nouvelle réponse, après discussion avec Fréderic :
"Salut Mikaël,

Je peux répondre à quelques unes des questions que tu m'as envoyées.

"En effet pour les étoiles qui ne sont pas de la séquence principale, on ne peut pas appliquer une formule du type L propto Massek. On a un autre moyen de déterminer la luminosité de ces étoiles en fonction des paramètres fournis ?"

Oui, je vais te fournir des modèles de paquets d'étoiles, semblables aux modèles d'amas globulaires (de l'ordre du million d'étoiles). Dedans tu auras entre autre la luminosité de chaque étoile et les magnitudes absolues. Donc pour avoir la luminosité d'un paquet constitué de ces étoiles, il faudra juste sommer, comme pour les amas.

"Avec Fux, on a seulement la position et la vitesse des paquets d'étoiles. Dans le modèle de galaxie final pour le projet RTIGE, tu penses qu'on aura la masse des paquets ?"

Oui, avec le catalogue que je vais fabriquer, tu auras ça, car on va avoir la masse de chaque étoile d'un paquet, et donc il suffira de sommer.

En fait, je vais ajouter des choses au modèle de Fux que je t'ai envoyé hier. Je vais ajouter une colonne masse et une colonne luminosité, et une colonne "modèle". La masse est la somme des masses des étoiles contenues dans le paquet (idem pour la luminosité). Dans la colonne "modèle", tu sauras quel modèle de paquet utiliser (cela dépend de la position du paquet dans la Galaxie).

Normalement avec ça on aura tout. Les données seront donc organisées de la même manière que pour les amas globulaires : un catalogue de positions de "paquets d'étoiles" et des modèles de paquets à distribuer à ceux-ci. Pour les amas ouverts, ça sera la même chose. L'avantage de ce système, c'est que si on change les modèles ou bien si on ajoute des paquets, cela marchera toujours.

"Donc en utilisant Pegase-HR comme le propose Francoise ?" Pour les modèles de paquets donc, oui il faut se décider. Je viens de discuter avec Carine Babusiaux. Elle me dit que le modèle de Besançon (différent du pegase-HR) n'est pas bon pour le bulbe et le halo de la Galaxie, mais qu'il convient très bien pour le disque. 1) pour le halo et le bulbe : il s'agit d'étoiles très vieilles, qui sont nées au tout début de la Galaxie, très rapidement. Elles ont donc toutes à peu près le même âge, et donc a peu près la même couleur. Ce n'est donc pas nécessaire de trop se creuser la tête pour elles. Il doit y avoir moyen de simplifier 1 seul modèle suffit. Je vais y réfléchir. 2) pour le disque : dans Besançon, ils le divisent en 8 étages de disque, le disque épais correspondant surtout au 8 ème étage. J'ai moyen dans le formulaire en ligne de ce modèle de simuler des étoiles dans ces différents étages. Il suffit que je découpe le disque du modèle de Fux de la même manière, et j'aurai ainsi 8 modèles selon que le paquet se situe dans l'un de ces étages.

"La première méthode semble la plus ouverte. on pourra toujours retomber sur la 2nde à partir des fluctuations en pré-générant 4 ou 5 modèles." Donc ça oui, il faut voir ce qui est le plus pratique et le

plus ouvert. Je peux pour l'instant attribuer à chaque paquet l'un des 9 modèles dont j'ai parlé plus haut, et te donner la fluctuation à apporter sur les paramètres. Dans ce système, on pourra toujours ajouter des modèles plus tard. Après je me dis, est-ce que c'est vraiment la peine d'avoir une si grande précision dans ces fluctuations ? est-ce que le spectateur verra vraiment une différence s'il manque une étoile A dans le paquet de gauche par rapport au paquet de droite ? peut-être que prendre 4 modèles différents pour un étage de disque suffit pour générer un peu d'aléatoire.

"Avec le modèle de galaxie Françoise, on avait du code source en fortran pour calculer les rayons des kernel pour chaque particule. Mais le code nous retournait une erreur. Donc pour mes tests j'ai fait un calcul de voisinage basique sans fonction de poids. Dans un premier temps on a besoin seulement du rayon du kernel et d'une fonction de poids, car on calculera uniquement des densités d'étoiles sur une grille. Le tirage des positions d'étoiles servira quand on zoomera beaucoup plus sur la galaxie."

Ca par contre je ne sais pas trop quoi en dire.

Voilà, avec ce système on est en mesure de faire des mises-à-jour si les connaissances et les modèles évoluent, sans changer le code. Si le modèle de paquets change, il suffit de remplacer les fichiers Si le modèle de galaxie (Fux) change (et il changera : Françoise Combes, Carine Babusiaux et Paola Di Matteo travaillent sur un nouveau modèle), ça n'altèrera pas le découpage des étages du disque. Pour les amas ouverts, je détecte les bras spiraux pour les positionner dessus. Si le modèle de bras change, je pourrai toujours les détecter."

Je retourne aux amas ouverts.

18 mai 2011
Voilà l'idée de Noël Robichon pour simuler les paquets, avec mes mots :

L'idée c'est d'avoir un diagramme H-R théorique qui contient toutes les étoiles de tous âges. Puis de le quadriller (faire un histogramme 2D) avec une certaine résolution.

Ensuite pour chaque sous-système du disque considéré, on connaît son âge moyen, et donc il faut vider les cases du diagramme H-R qui ne peuvent contenir d'étoiles de cet âge-là. Puis simuler la distribution d'étoiles d'un paquet situé dans ce sous-système à partir des cases non-vides.

J'ai donc besoin de l'âge des étoiles en fonction de leur position dans le disque. Est-ce qu'il y a une variation avec l'altitude par rapport au plan du disque ? selon la distance au centre galactique ? Peut-on prendre le même âge pour toutes les étoiles du bulbe et du halo ?

19 mai 2011
Misha Haywood me dit que oui, l'âge des étoiles varie avec l'altitude, mais qu'on ne sait pas s'il varie avec la distance au centre galactique :
"La façon qui est utilisée de représenter ça, c'est de décrire la répartition en Z avec des exponentielles d'échelles de hauteur plus grandes pour les population plus âgées. par exemple, pour du vieux disque, la loi peut être décrite par exp(-Z/hZ) avec hZ l'échelle de hauteur, hZ=300pc pour du disque jeune (¡1Gyr), on plutôt hZ=100pc. entre les deux, disons que tu peux garder 300pc entre 3-4Gyr et 10Gyr, et entre 0 et 3-4, tu interpole entre 100 et 300pc. ça ne te donne pas directement un âge moyen en fonction de Z, qui dépend encore d'autres paramètres, mais ça peut fournir une première approximation."

J'ai donc la relation :

$$N(Z, age) = \exp -Z/h_Z(age) = \exp -Z/(100(pc/Gyr) * age) \quad (8)$$

avec Z en pc et age en Gyr. Je prends une simple proportionnalité entre h_Z et l'âge des paquets.

Je regarde la distribution des Z des particules du disque dans le modèle de Fux :

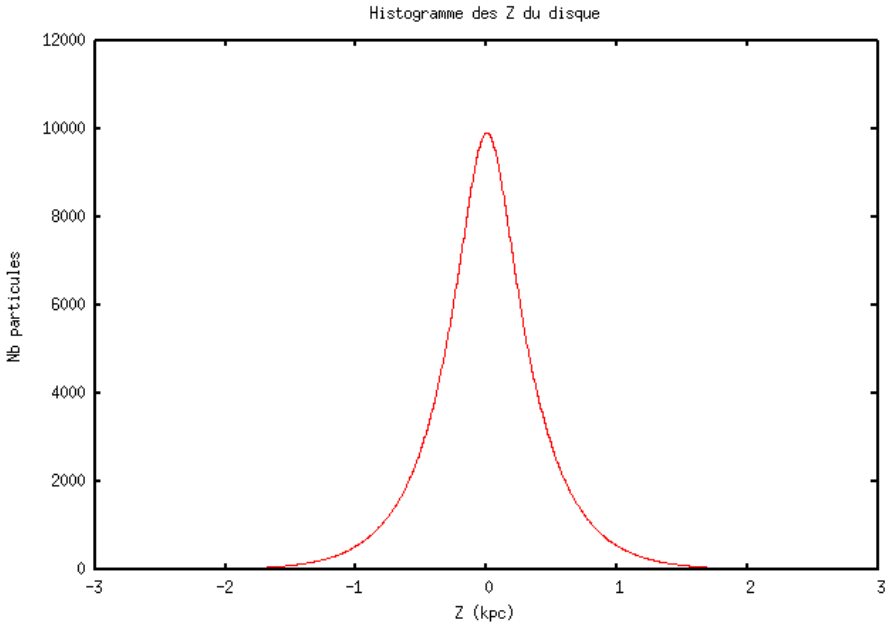


FIGURE 1 –

En appliquant la relation (12) aux âges :
(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.,1.5,2.,3.,4.,5.,6.,7.,8.,9.,10.,11.,12.) Gyr (de 0.1 à 0.2 Gyr, de 0.2 à 0.3 Gyr ...),
j'obtiens les histogrammes de la *figure* 2. Les courbes, de bas en haut, correspondent à des âges croissants. J'ai pris 500 boîtes, pour Z entre 0 et 3 kpc. Les histogrammes sont normalisés à 1, car il n'y a pas de coefficient devant l'exponentielle de la relation (12). Celui-ci est le nombre total d'étoiles de chaque âge présentes dans le disque pour $Z \in [0:3]$ kpc. Si l'on considère que le nombre total d'étoiles est le même pour tout $Z \in [0:3]$ kpc et pour chaque tranche d'âge de 0-1 Gyr, jusque 12-13 Gyr, Misha Haywood me dit que c'est une description raisonnable, au vu du manque de connaissances que l'on a là-dessus.

Dans le disque de Fux (fin et épais mélangés), j'ai $N_P = 1335167$ paquets d'étoiles soit $\sim 1.3 \times 10^6$ paquets d'étoiles. Tous les paquets ont la même masse. Comme ils contiennent un grand nombre d'étoiles, je peux considérer qu'ils contiennent aussi le même nombre d'étoiles. Il y a $\sim 2 \times 10^{11}$ étoiles. Donc on a $N_{*/P} \sim 1.5 \times 10^6$ étoiles par paquets.

Je dois donc répartir $N_P/2.$ paquets d'étoiles dans ces 22 tranches d'âge (je prends que la partie $Z >= 0$ du disque pour l'instant. Pour une tranche d'âge i donnée, soit N_{P_i} le nombre de paquets à distribuer dans les tranches d'altitude. Il faut alors normaliser la relation (12) comme suit pour avoir le nombre $N_P(Z)$ de paquets dans la tranche d'altitude Z pour l'âge i.

$$N_P(Z) = \frac{N_{P_i}}{\sum_j \exp\left(-Z_j/(100(pc/Gyr) * age_i)\right)} \exp\left(-Z/(100(pc/Gyr) * age_i\right) \tag{9}$$

La *figure* 3 donne les histogrammes calculés avec ces poids.

Ce graphe me donne directement les abondances de paquets d'étoiles de chaque classe d'âge pour chaque classe d'altitude dans le disque. Je peux visualiser ces résultats sous forme d'un histogramme à 2D (soit une image) (voir *figure* 4 non normalisé, et *figure* 5 normalisé). En abscisses, l'altitude en kpc, en ordonnées, l'âge en Gyr (l'axe des ordonnées à été étiré par un facteur 10 pour mieux voir, car il y a 22 âges seulement).

20 mai 2011
Avec les poids (voir *figure* 5) :

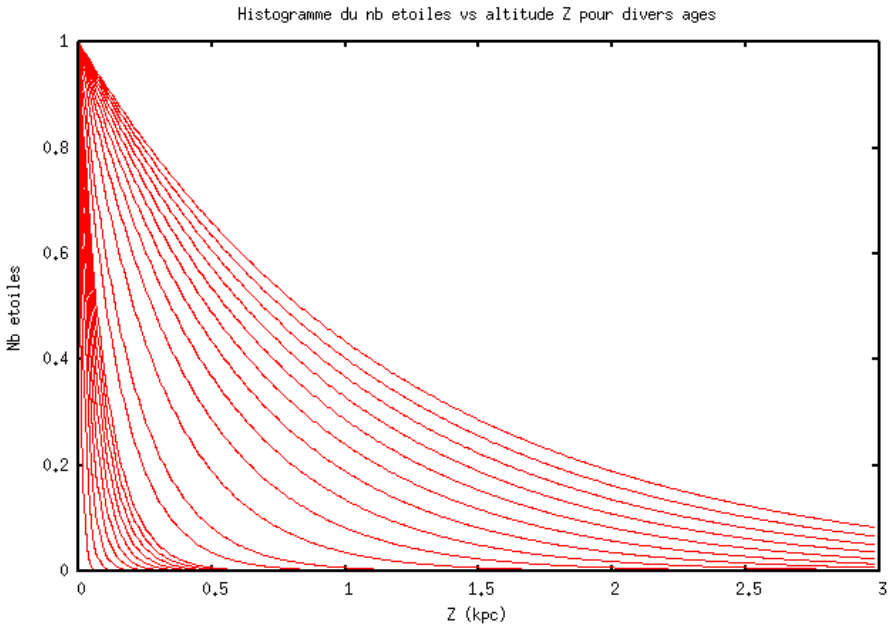


FIGURE 2 –

La cassure visible peu avant la moitié de l'axe des ordonnées est dû au fait qu'on a pris des classes d'âges de tailles différentes, en particulier entre 1 et 2 Gyr. La redistribution des paquets d'étoiles selon les âges tient compte de ça. Lorsque je fais la somme des valeurs de l'image, j'obtiens 667683.5 paquets, soit la moitié des paquets d'étoiles du disque (on a pris $Z >= 0$).

Il reste maintenant à répartir les paquets d'étoiles présents dans le fichier selon leur Z, et leur attribuer un âge, en suivant cet histogramme 2D. En considérant le disque comme étant symétrique par rapport au plan, on effectuera ce processus également pour $Z < 0$.

Pour avoir un nombre de paquets entiers des deux côtés du plan, j'enlève le paquet le plus proche du plan.

2 août 2011

Comme je l'ai fait pour les amas globulaires, je ne vais pas simuler des paquets d'étoiles mais fournir les outils pour le faire à RSAcosmos. Je vais utiliser la méthode proposée par Noël Robichon :

1. Découper la Galaxie en sous-systèmes : halo, disque mince, disque épais, centre galactique. Voir ensuite s'il faut faire plus de détails.

2. Construire un diagramme H-R théorique (M_V, $(B-V)$) qui contient tous les types d'étoiles possibles à partir du modèle de Besançon calculé pour un grand nombre d'étoiles, représentatif d'un sous-système de la Galaxie. Pour les tests j'ai fait une requête sur le site du modèle de Besançon en demandant un échantillon de 0 à 8kpc du Soleil, centré sur la position l=0, b=0, et pour un angle solide de 1 degré carré, sans extinction. J'ai pris tous les âges, tous les types spectraux de O0 à DA5, et m_V entre 10 et 16.

3. Faire un histogramme 2D normalisé à 1 de ce diagramme H-R théorique avec des boîtes de taille suffisamment petite pour bien représenter tous les types d'étoiles présentes. J'ai fait un diagramme HR de 300 boîtes par 300 boîtes : *figure* 6. Le diagramme est un peu écrasé en $(B-V)$, car les échelles ne sont pas les mêmes. Du coup les types d'étoiles seront moins variés pour les étoiles chaudes. Je vais plutôt faire l'histogramme de $\log((B-V)+2)$ pour l'axe des abscisses : *figure* 7. Il faudra tirer des valeurs non pas de $(B-V)$ mais de $K_1 = \log((B-V)+2)$ pour simuler des étoiles. J'appelle K_2 les valeurs de M_V et dN les valeurs de l'histogramme.

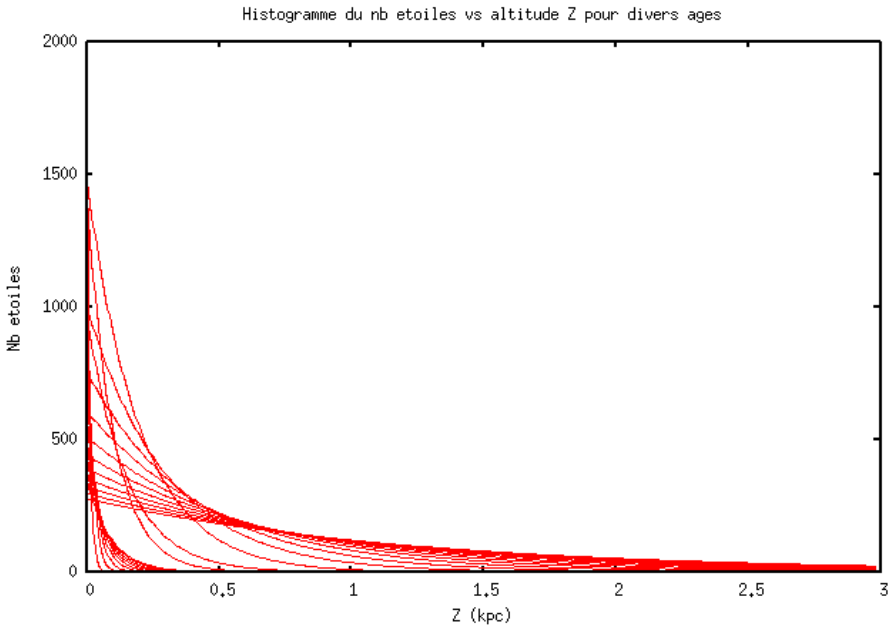


FIGURE 3 –

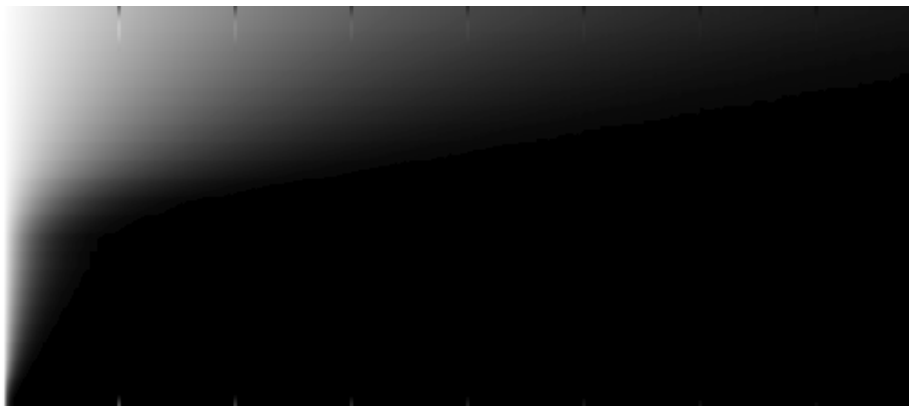


FIGURE 4 –

4. Tirer des paramètres stellaires avec l'algorithme suivant :

$i = 0$
while $i < nb_{etoiles}$:
k_1 =valeur tirée aléatoirement selon une distribution uniforme entre 0 et $K_{1_{max}}$.
k_2 =valeur tirée aléatoirement selon une distribution uniforme entre 0 et $M_{V_{max}}$.
dn =valeur tirée aléatoirement selon une distribution uniforme entre 0 et dN_{max}.
dN_{tmp} =valeur de dN pour le couple (K_1, K_2) le plus proche de (k_1, k_2) dans l'histogramme.
if $dn <= dN_{tmp}$:
On retient les valeurs k_1 et k_2, créant ainsi une nouvelle étoile.
On retient également les valeurs des autres paramètres donnés par
le modèle pour le couple (K_1, K_2).
$i+ = 1$

Attention : Le $i+ = 1$ doit être dans le **if**, afin que si le test n'est pas passé, une nouvelle étoile soit tirée, car il faut $nb_{etoiles}$.

5. Tirer uniformément des positions pour les étoiles créées dans une sphère de rayon R_{max}, centrée



FIGURE 5 –

sur la particule concernée dans le modèle de Fux. On peut aussi tirer des étoiles dans une portion de la sphère (voir plus loin). R_{max} est la distance moyenne des particules les plus proches dans chaque direction de la particule concernée. Il devrait ainsi il y avoir un peu de recouvrement avec les sphères des particules proches. Voici l'algorithme :

$i = 0$

while $i < nb_{etoiles}$:

$k =$valeur tirée aléatoirement selon une distribution uniforme entre 0 et 1.

$R = k^2 R_{max}$

$S =$valeur tirée aléatoirement selon une distribution uniforme entre -1 et 1.

$C = R \times \sqrt{1 - S^2}$

$\phi =$valeur tirée aléatoirement selon une distribution uniforme entre $-\pi$ et π.

$x[i] = C \times \cos\phi$

$y[i] = C \times \sin\phi$

$z[i] = R \times S$

$i+ = 1$



FIGURE 6 –



FIGURE 7 –

On peut aussi faire les deux algorithmes à la fois : pour chaque position x,y,z tirée, tirer un indice de couleur et une magnitude :

$i = 0$
while $i < nb_{etoiles}$:
k =valeur tirée aléatoirement selon une distribution uniforme entre 0 et 1.
$R = k^2 R_{max}$
S =valeur tirée aléatoirement selon une distribution uniforme entre -1 et 1.
$C = R \times \sqrt{1 - S^2}$
ϕ =valeur tirée aléatoirement selon une distribution uniforme entre $-\pi$ et π.
$x[i] = C \times \cos\phi$
$y[i] = C \times \sin\phi$
$z[i] = R \times S$
$j = 0$
while $j < 1$:
k_1 =valeur tirée aléatoirement selon une distribution uniforme entre 0 et $K1_{max}$.
k_2 =valeur tirée aléatoirement selon une distribution uniforme entre 0 et $M_{V_{max}}$.
dn =valeur tirée aléatoirement selon une distribution uniforme entre 0 et dN_{max}.
dN_{tmp} =valeur de dN pour le couple (K_1, K_2) le plus proche de (k_1, k_2) dans l'histogramme.
if $dn <= dN_{tmp}$:
On retient les valeurs k_1 et k_2, créant ainsi une nouvelle étoile.
On retient également les valeurs des autres paramètres donnés par
le modèle pour le couple (K_1, K_2).
$j+ = 1$
$i+ = 1$

La boucle **while** sur j est là pour tirer des paramètres pour l'étoile i jusqu'à ce que le test sur dn soit passé.

Enfin, pour tirer des étoiles dans une portion de la boule limitée par R_1, R_2, θ_1, θ_2, ϕ_1 et ϕ_2, il faut :

1. Calculer le nombre d'étoiles nb_C présentes dans la coquille de rayons R_1 et R_2. Soit V_S le volume de la sphère de rayon R_{max}, et V_C le volume de la coquille, alors :

$$nb_C = \frac{nb_{etoiles} V_C}{V_S} = \frac{nb_{etoiles}(R_2^3 - R_1^3)}{R_{max}^3} \tag{10}$$

2. Faire l'algorithme suivant :

$i = 0$
while $i < nb_C$:
  k =valeur tirée aléatoirement selon une distribution uniforme entre 0 et 1.
  $R = k^2 R_{max}$
  S =valeur tirée aléatoirement selon une distribution uniforme entre -1 et 1.
  $C = R \times \sqrt{1 - S^2}$
  ϕ =valeur tirée aléatoirement selon une distribution uniforme entre $-\pi$ et π.
  if $\theta_1 < \arcsin(S) < \theta_2$ and $\phi_1 < \phi < \phi_2$ and $R_1 < R < R_2$:
    $x[i] = C \times \cos\phi$
    $y[i] = C \times \sin\phi$
    $z[i] = R \times S$
    $j = 0$
    while $j < 1$:
      k_1 =valeur tirée aléatoirement selon une distribution uniforme entre 0 et $K_{1_{max}}$.
      k_2 =valeur tirée aléatoirement selon une distribution uniforme entre 0 et $M_{V_{max}}$.
      dn =valeur tirée aléatoirement selon une distribution uniforme entre 0 et dN_{max}.
      dN_{tmp} =valeur de dN pour le couple (K_1, K_2) le plus proche de (k_1, k_2)
      dans l'histogramme.
      if $dn <= dN_{tmp}$:
        On retient les valeurs k_1 et k_2, créant ainsi une nouvelle étoile.
        On retient également les valeurs des autres paramètres donnés par
        le modèle pour le couple (K_1, K_2).
        $j+ = 1$
  $i+ = 1$

Attention : Le $i+ = 1$ doit être dans le **premier while**, car les n étoiles présentes sur la surface de sphère de rayon r ne sont pas toutes sur la portion de surface limitée par θ_1, θ_2, ϕ_1 et ϕ_2. Les étoiles ne passant pas le test sur θ, ϕ et R font partie des $nb_{etoiles}$ mais ne sont pas dans le champ de vision choisi.

Découpage de la Galaxie Dans le formulaire du modèle de Besançon, on peut choisir de simuler ou non le disque, la population intermédiaire (que je suppose être le disque épais), le halo et le bulbe. Donc, en demandant 5 degrés carrés autour de la direction du centre galactique, pour des magnitudes apparentes en bande V de 10 à 20, et des distances entre 0 et 8 kpc du Soleil, en sélectionnant le disque fin, puis le disque épais, puis le halo, puis le bulbe, je devrais avoir quatre diagrammes H-R de référence.

Pour passer de l,b,d à X_S,Y_S,Z_S, coordonnées cartésiennes centrées sur le Soleil, il faut utiliser les transformations de sphériques vers cartésiennes, soit :

$$X_S = d * cos(l) * cos(b) \tag{11}$$

$$Y_S = d * sin(l) * cos(b) \tag{12}$$

$$Z_S = d * sin(b) \tag{13}$$

L'axe Ys pointe dans la direction de rotation de la Galaxie, l'axe X_S pointe vers le centre galactique, et l'axe Z_S est perpendiculaire au plan (X_S, Y_S) de manière à ce que le repère soit direct.

19 Août 2011

La dernière fois que l'on s'est vus, Frédéric m'a dit de plutôt travailler en V-I. On a aussi discuté du recouvrement des régions autour des particules de Fux. Plutôt que des sphères qui laissent des trous, il faudrait des cubes. Cependant, il faut transformer un peu le modèle de Fux : pour l'instant , on a des particules contenant un nombre identique d'étoiles (même masse totale), et qui sont du coup séparées par diverses distances. Il faudrait des particules espacées selon une grille, mais contenant divers nombres d'étoiles. Je dois trouver un algorithme qui calcule ça. Enfin, pour ce qui est de l'âge des étoiles, comme on n'utilise pas d'isochrones, on ne peut pour l'instant pas faire vieillir ou rajeunir les étoiles. La solution, c'est de calculer des diagrammes H-R pour différents âges, et de fournir des cubes composés d'histogrammes de diagrammes H-R.

Je me mets aux téléchargements de modèles de Besançon :

- Disque mince :
  Je demande au serveur les paramètres suivants : distance de 4 à 4.2kpc, cible (l=0, b=0) ; solid angle 5 deg2 ; Av=0.0mag/kpc ; types de O0 à DA5 toutes les classes de luminosité ; populations 01 à 07 inclus ; V entre 10 et 99.
- Disque épais :
  Je demande au serveur les paramètres suivants : distance de 4 à 5kpc, cible (l=0, b=0) ; solid angle 5 deg2 ; Av=0.0mag/kpc ; types de O0 à DA5 toutes les classes de luminosité ; population 08 uniquement ; V entre 10 et 99.
- Halo :
  Je demande au serveur les paramètres suivants : distance de 11 à 12kpc, cible (l=0, b=0) ; solid angle 5 deg2 ; Av=0.0mag/kpc ; types de O0 à DA5 toutes les classes de luminosité ; population 09 uniquement ; V entre 10 et 99.
- Bulbe :
  Je demande au serveur les paramètres suivants : distance de 8 à 8.2kpc, cible (l=0, b=0) ; solid angle 5 deg2 ; Av=0.0mag/kpc ; types de O0 à DA5 toutes les classes de luminosité ; population 10 uniquement ; V entre 10 et 99.

22 Août 2011

J'ai pu télécharger les données pour le disque épais et le halo vendredi, mais le serveur ne répond plus ce matin. Je ne peux pas télécharger les données pour le disque mince et le bulbe. Je vais commencer à travailler sur ce que j'ai. Je construit un diagramme de Hess pour ces deux parties de la Galaxie avec le programme 'RTIGE_SP_Hess_diagrams.py'. Attention, l'image crée par le programme est renversée et tournée. Pour avoir la bonne visualisation, il faut faire un miroir horizontal et une rotation anti-horaire de 90°. Les diagrammes de Hess obtenus sont présentés sur les *figures* 8 et 9. J'ai modifié la luminosité et le contraste des images pour bien voir les points. Je vais utiliser des isochrones pour avoir des diagrammes de Hess correspondant à différents âges. Il faut choisir des métallicités.

Pour le disque mince je reprends la figure 1 de l'article de Magrini et al., où ils donnent la métallicité d'amas ouverts en fonction du rayon galactique. (voir *figure* 10).

Dans Navarro et al. 2011, on donne $-0.7 < [Fe/H] < 0.2$ pour le disque mince au voisinage solaire. Je vais prendre 3 ou 4 valeurs différentes selon R. Pour coller à peu près à la figure de Magrini et al. 2009, je vais prendre 4 intervalles de rayon et une valeur de [Fe/H] pour chaque : entre 0 et 3 kpc, $[Fe/H] = 0.5$; entre 3 et 8 kpc, $[Fe/H] = 0$; entre 8 et 13 kpc, $[Fe/H] = -0.4$; entre 13 et 20 kpc, $[Fe/H] = -0.5$.

Voilà comment je passe de [Fe/H] (donnés par Harris 1997) à Z (méthode décrite dans Lebreton 1997) :

$$[Fe/H] \approx \log Z/X - \log (Z/X)_{\odot} \tag{14}$$

$$Z \approx X \times 10^{[Fe/H]+\log (Z/X)_{\odot}} \tag{15}$$

$$Z \approx \frac{(1-Y) \times 10^{[Fe/H]+\log (Z/X)_{\odot}}}{1 + 10^{[Fe/H]+\log (Z/X)_{\odot}}} \tag{16}$$

Par définition :

$$X + Y + Z = 1 \tag{17}$$



FIGURE 8 –

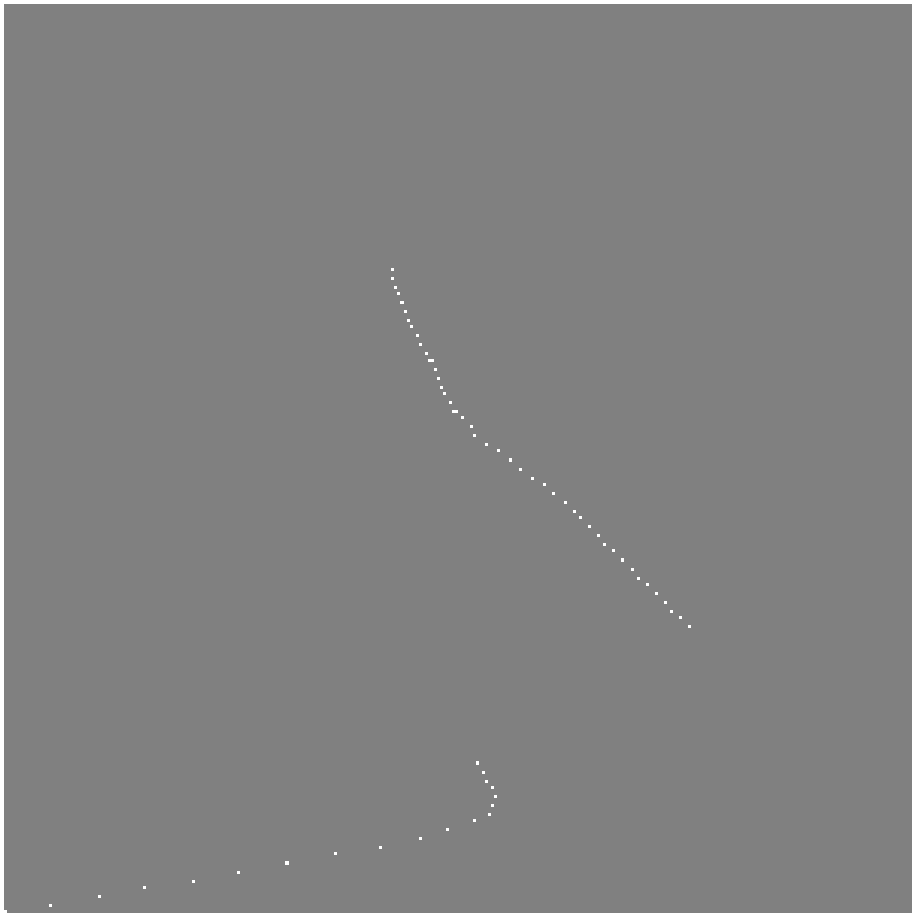


FIGURE 9 –

Avec X l'abondance d'hydrogène, Y l'abondance d'hélium, et Z l'abondance de tous les éléments plus lourds que l'hélium. Il est raisonnable de prendre le Y solaire au vu de toutes les approximations que j'ai faites. Les valeurs solaires sont : $X_\odot = 0.7381$, $Y_\odot = 0.2485$, $Z_\odot = 0.0134$ (Asplund 2009) J'avais déjà écrit une fonction en python pour faire ce calcul, dans la librairie RTIGEutil. On a donc ces 4 métallicités Z : entre 0 et 3 kpc, $Z = 0.04$; entre 3 et 8 kpc, $Z = 0.01$; entre 8 et 13 kpc, $Z = 0.005$; entre 13 et 20 kpc, $Z = 0.004$.

Pour le disque épais Dans Navarro et al. 2011 (figure 1 de l'article), on donne $-1.3 < [Fe/H] < -0.2$ pour le disque épais au voisinage solaire. Je vais prendre la moyenne, soit $[Fe/H] = -0.75$, donc $Z = 0.002$.

Pour le Halo Dans Navarro et al. 2011, on donne $[Fe/H] < -1.5$ Je vais prendre $[Fe/H] = -1.5$ soit $Z = 0.0004$. Je vais prendre Y=0.26, comme pour les amas globulaires.

isochrones Je télécharge les fichiers d'isochrones sur le serveur YZVAR. Paramètres : $\log(age/yr)_max = 10.30$ et $\Delta log(age) = 0.01$. Le serveur ne fonctionne pas. Je vais prendre les fichiers d'isochrones proposés sur le site : Z=0.001, Z=0.0001, Z=0.002, Z=0.004, Z=0.0004, Z=0.008, Z=0.017, Z=0.040 avec Y=0.26.
Pour le disque mince je vais donc prendre : $Z = 0.04$ entre 0 et 3 kpc, $Z = 0.017$ entre 3 et 8 kpc, et $Z = 0.004$ au-delà de 8kpc.
Pour le disque épais je vais prendre $Z = 0.002$.
Pour le halo je vais prendre $Z = 0.0004$

26 août 2011
Je m'attèle au découpage de la galaxie. Je plotte la fonction $ln(\phi(z))$, normalisé à , pour les étoiles du disque du modèle de Fux. C'est le logarithme népérien de l'histogramme des altitudes. (voir *figure* 11) . Le disque épais est censé avoir un histogramme lus large et moins haut que le disque mince. En faisant l'histogramme des deux disques mélangés, on devrait donc voir une cassure. On n'en voit pas, il n'y a donc pas de disque épais dans ce modèle.

Dans le fichier sphero.dat, il y a le bulbe (R¡3kpc) et le halo. Je prends un âge de 12Gyr pour le bulbe et 13 Gyr pour le halo. Mélanie Guittet me dit que pour le halo on a $[Fe/H] = -1.5$ avec un écart type de 0.3. Je vais prendre 3 valeurs : $[Fe/H] = -1.5$, $[Fe/H] = -1.2$ et $[Fe/H] = -1.8$, et les répartir de façon gaussienne aux particules du bulbe et du halo. Ces trois valeurs correspondent à $Z = 0.0004$, $Z = 0.0008$ et $Z = 0.0002$ en prenant $Y = Y_\odot$.

Quand j'aurai réussi à fabriquer un disque épais, je donnerai l'âge 12 Gyr et un $[Fe/H]$ moyen de -0.65 (Z=0.003). Mélanie Guittet m'a donné cet âge et un intervalle[-0,5 ;-0,8] pour $[Fe/H]$.

Pour le disque mince, je vais prendre 3 métallicités différentes selon R : pour $0 < R < 3$ kpc : [Fe/H]=0.5 (Z=0.03), pour $3 < R < 8$ kpc : [Fe/H]=0 (Z=0.01), pour $R > 8$ kpc : [Fe/H]=-0.5 (Z=0.004). Pour les âges, je vais appliquer un gradient vertical avec une dizaine de valeurs, et les attribuer aux particules.

Cela fait donc 7 métallicités différentes à simuler, soit 7 fichiers d'isochrones à télécharger (http ://stev.oapd.inaf
bin/cmd_2.3). Je les mets dans '/Users/yannickboissel/RTIGE/data/IsochronesGirardi/Disc_stars'.
Pour le disque mince, Z=0.04 est au-delà de la limite du calculateur, je prends 0.03 à la place. Je modifie et lance 'RTIGE_OC_isochrones.py' pour créer les fichiers RTIGE associés.

30 août 2011
J'ai eu un échange de mails avec Mikaël. Le voici :

moi : "Hello,

Je me pose une question à laquelle toi seul peut répondre :
Pour simuler la galaxie actuelle, je vais te donner un modèle de galaxie modifié (Fux ou Combes), avec pour chaque particule un âge (une dizaine de valeurs différentes) et une métallicité (7 valeurs

différentes).

Je vais te donner aussi 7 histogrammes de diagramme H-R (1 par métallicité), indiquant la proportion d'étoiles à tirer de chaque type pour un nombre donné d'étoiles.

Enfin, 7 fichiers d'isochrones (comme pour les amas globulaires) afin de récupérer pour chaque case de ces diagrammes H-R les autres paramètres stellaires.

Voilà la question : Pour faire vieillir ou rajeunir la galaxie, préfères-tu : 1) Pour chaque métalicité, 1 diagramme H-R correspondant à l'âge actuel. On tire les étoiles actuelles et on les fait vieillir ou rajeunir avec les fichiers d'isochrones.

2) Pour chaque métallicité, 1 cube de 354 diagrammes H-R correspondant aux 354 âges des fichiers d'isochrones. On peut directement tirer les étoiles pour n'importe quel âge, et on récupère les autres paramètres dans les fichiers d'isochrones. Mais on n'a pas besoin des isochrones pour vieillir les étoiles.

Dans le premier cas, on a peu de données sur le disque, mais il faut toujours partir des étoiles tirées à l'âge actuel de la galaxie pour simuler des étoiles d'un autre âge. Dans le second cas on a plus de données sur le disque, mais on peut directement tirer des étoiles à l'âge qu'on veut.

Qu'en penses-tu en termes de temps de calcul ?

Yannick"

Mikaël : "Salut Yannick,

Je n'ai pas trop regardé la problématique du changement d'âge de la galaxie. Et je pense que je ne suis pas près de travailler dessus, car il y a encore beaucoup de travail pour la galaxie à l'âge actuel. Donc pour l'instant tu peux te contenter de travailler sur les données de l'âge actuel uniquement.

Cependant, à terme, la 2e solution sera peut-être plus simple car la première implique de gérer la mort et la naissance des étoiles entre 2 âges, non ? "

moi : " Hello, Eh bien, il y a du pour et du contre dans les deux cas :

dans le cas 1), c'est utile lorsqu'on doit tirer des étoiles : on les fait viellir à partir d'un âge donné. Si on utilisait plusieurs diagrammes H-R, on ferait viellir des étoiles qui changent de place ! (en effet on devrait tirer leur position pour chaque âge). Donc dans le cas où l'on discerne les étoiles, les faire vieillir avec les isochrones est plus intéressant.

dans le cas 2) : Lorsqu'on voit la galaxie de loin, on ne voit pas les étoiles. Les isochrones sont alors inutiles. On veut de toute façon que la luminosité globale et couleur moyenne d'un paquet de particules de modèle. Pour ça, des diagrammes H-R suffisent pour simuler la galaxie à n'importe quel âge avec très peu de calcul (on ne tire pas d'étoiles).

disons que les fichiers d'isochrones permettent d'afficher une évolution entre deux âges, et que les diagrammes H-R permettent d'afficher un ensemble d'étoile directement à un âge donné (mais pas de faire évoluer).

Ce que je pense , c'est que si je te donne les cubes et les isochrones, tu peux faire ce que tu veux dans tous les cas de figure. C'est peut-être plus simple :) " Je vais donc mettre dans une archive à la fois des cubes de diagrammes H-R et des fichiers d'isochrones, afin d'avoir le plus de liberté d'action possible. J'ai déjà téléchargé tous les fichiers d'isochrones dont j'ai besoin le 26 août. J'avais commencé à écrire le programme qui fabrique les cubes de diagrammes H-R à partir des fichiers d'isochrones : 'RTIGE_HR_cubes.py'.

J'ai fait des cubes de 100 par 100 par 354 (âges). Log(L) en ordonnées et log(Te) en abscisses. Ils sont dans le dossier '/Users/yannickboissel/RTIGE/data/HR/'. Les fichiers font 7.1 Mo chacun. Les valeurs des cases sont divisées par le total des valeurs des cases et multipliées par 1000.

31 août 2011

Je n'ai pas de naines blanches dans les isochrones. Il faut que je trouve une solution à ça.

Pour tirer les étoiles : faut-il considérer un nombre identique d'étoiles par particules ou bien une masse identique ?

Comment faire pour la variation de l'âge avec l'altitude ?

1 septembre 2011

Pour tirer des masses plutôt que des nombres d'étoiles, voilà ce qu'il faut faire : 1) Fabriquer une image de même taille qu'un diagramme HR donnant la masse pour chaque couple $\log(L_V), \log(Te)$, pour chaque métallicité. Cela fait 7 images à mettre en première slice des cubes. Pour une métallicité donnée, un même couple $\log(L_V), \log(Te)$ correspond à la même masse quelque soit l'âge, normalement. (sauf perte de masse pour les géantes rouges, que je vais ignorer). Une masse est donnée également pour les positions impossibles, mais on ne les utilisera pas, donc ça ne gêne pas. La *figure* 13 montre une image de diagramme HR pour un âge quelconque, et la carte de masse correspondante. (j'ai modifié le programme 'RTIGE_HR_cubes.py')

Je modifie l'algorithme de tirage d'étoiles :

$m = 0$
$i = 0$
while $m < masse_{particule}$:
  $x[i]$ =valeur tirée aléatoirement selon une distribution uniforme entre 0 et a.
  $y[i]$ =valeur tirée aléatoirement selon une distribution uniforme entre 0 et a.
  $z[i]$ =valeur tirée aléatoirement selon une distribution uniforme entre 0 et a.
  $j = 0$
  while $j < 1$:
    $\log(l_V/L_\odot)$ =valeur tirée aléatoirement selon une distribution uniforme
    entre $\log(L_V/L_\odot)_{min}$ et $\log(L_V/L_\odot)_{max}$.
    $\log(te)$ =valeur tirée aléatoirement selon une distribution uniforme
    entre $\log(Te_{min})$ et $\log(Te_{max})$.
    dn =valeur tirée aléatoirement selon une distribution uniforme entre 0 et dN_{max}.
    dN_{tmp} =valeur de dN pour le couple $(\log(L_V/L_\odot), \log(Te))$ le plus proche de $(\log(l_V/L_\odot), \log(te))$
    dans l'histogramme.
    m_{tmp} = masse pour le couple $(\log(L_V/L_\odot), \log(Te))$ ainsi repéré, prise dans l'image de masse.
    if $dn <= dN_{tmp}$:
      On retient les valeurs $\log(l_V/L_\odot)$ et $\log(te)$, créant ainsi une nouvelle étoile.
      On retient également les valeurs des autres paramètres donnés par
      le modèle pour le couple $(\log(L_V/L_\odot), \log(Te))$.
      $j+ = 1$
  $i+ = 1$
  $m+ = m_{tmp}$

Maintenant il faut que je résolve le problème des étoiles qui meurent. Logiquement, le long d'un cube, avant normalisation, le nombre d'étoiles (la somme de toutes les cases) doit diminuer. Je dois donc pouvoir savoir, en comparant deux âges dans un fichier d'isochrones, combien d'étoiles ont disparu entre deux âges, et comme je connais la masse maximale des étoiles existantes pour un âge donné, je sais quelle est la masse minimale de ces étoiles manquantes. Je sais donc si elles sont des trou noirs ou étoiles à neutrons, ou alors des naines blanches. Je dois donc être capable d'ajouter ces étoiles dans les fichiers d'isochrones. Ainsi, la fabrication des cubes de diagrammes H-R et des cartes de masse se fera sans encombre, et tiendra compte de ces étoiles. Je fais une copie du programme 'RTIGE_OC_isochrones.py' que je garde pour les amas ouverts. La copie s'appelle 'RTIGE_S_isochrones.py' avec 'S' pour Stars. Je renomme également le programme 'RTIGE_HR_cubes.py' en 'RTIGE_S_HR_cubes.py'.

5 septembre 2011

J'ai écrit l'algorithme qui ajoute les étoiles mortes aux isochrones. Je suis en train de le débugger.

7 septembre 2011

L'algorithme qui ajoute les naines blanches est fini et fonctionne ('RTIGE_S_isochrones.py'). Cependant il y a un problème : dans les fichiers d'isochrones, le nombre de lignes pour un âge donné ne représente pas le nombre relatif d'étoiles de chaque masse. Cette quantité doit être calculée à partir

de la dernière colonne du fichier "int_IMF".

Pour le choix de l'IMF, j'avais pris le choix par défaut, Chabrier 2001 lognormal. Misha Haywood me conseille plutôt de prendre Kroupa 1998 corrected for binaries, plus adapté aux étoiles de champ.

D'après l'aide sur le site CMD, on a :

$$int_{IMF}[i] = \int_0^{m_{[i]}} IMF(m)dm \tag{18}$$

Donc, le nombre relatif d'étoiles $dN[i]$ entre deux valeurs de masse $m[i-1]$, que l'on est obligé d'associer à $m[i]$ (il faut bien mettre une résolution en masse quelque part) est :

$$dN[i] = int_{IMF}[i] - int_{IMF}[i-1] \tag{19}$$

8 septembre 2011
J'ai fait des tests pour voir si $int_{IMF}[i] - int_{IMF}[i-1]$ était bien normalisée à 1, i.e, la somme fait bien 1. Cela dépasse de loin 1 si l'on considère $dN[i] = int_{IMF}[i] - 0$ pour $i = 0$. En fait, comme je ne connais pas la valeur de la distribution pour une masse nulle, je ne peux pas avoir de valeur de dN pour une masse de $0.15M_\odot$. Je dois donc commencer à $0.20M_\odot$. Pour $z = 0.006$ et une IMF de Kroupa corrigée des étoiles binaires, la somme fait 0.6355 sans la première masse. Cela veut dire que les étoiles de masse inférieure à $0.15M_\odot$ comptent pour $1 - 0.6355 \approx 0.36M_\odot$. Ainsi je retrouve les données manquantes.

12 septembre 2011
Je suis allé voir Carine Babusiaux puis Noël Robichon cet après-midi. J'ai discuté des isochrones avec Carine. C'est bien ce que j'avais compris : En soustrayant deux valeurs de int_{IMF} successives j'ai le nombre relatif d'étoiles, calculé de sorte que la masse totale des étoiles fasse $1M_\odot$. Donc, la somme des différences des valeurs de int_{IMF} doit faire un peu moins de 1. Elle ne fait pas 1, car on ne peut pas calculer le nombre relatif d'étoiles de la première masse donnée dans l'isochrone en faisant une différence de valeurs de int_{IMF}. Pour trouver cette valeur il suffit de soustraire la somme des différences à 1. En ce qui concerne l'âge des étoiles dans le disque, elle me dit que l'approximation que m'avait donné Misha Haywood (voir au début de ce journal de bord), soit en prenant une échelle de hauteur proportionnelle à l'âge, ça convient pour ce que l'on veut faire.
Pour l'âge des naines blanches, j'ai posé la question à Noël Robichon : il pense que les âges donnés par Chabrier et al. 2000 sont les âges des étoiles depuis qu'elles sont devenues naines banches, comme ce que j'avais compris quand j'ai travaillé sur les amas globulaires. Pas de changements à faire sur la méthode pour les naines blanches donc.
J'avais enlevé les âges supérieurs à 13 Gyr dans les fichiers 'WD##.dat', probablement parce que cela donnait des âges plus grands que l'âge de l'Univers. Je les remets au cas où on veuille faire vieillir la Galaxie. Je lance le programme 'RTIGE_evolutionary_tracks.py' pour corriger les fichiers 'RTIGE_2011_WD_evol_path_##.dat'.

13 septembre 2011
Afin de bien construire l'algorithme qui ajoute les étoiles mortes aux isochrones, j'ai tout réécrit il y a quelques jours dans le programme 'test10.py'. Maintenant il marche. J'ai ajouté les paramètres stellaires dans l'algorithme. C'est le programme 'RTIGE_S_isochrones.py'. Il tourne, mais c'est très lent. Après corrections d'encore d'autres bugs, le programme fonctionne (c'était lent parce que j'avais oublié de fermer des boucles). Mon ordinateur ne permet pas de faire tourner ce programme rapidement, alors je le lance sur Aquilon.

15 septembre 2011
Si dans 'RTIGE_S_isochrones.py' je prends $N_{tot} = 10000$ étoiles je ne peux pas convertir les digrammes HR en int16. Si je prends $N_{tot} = 1000$, je n'ai pas de géantes sur les diagrammes. Je vais prendre $100 * \log$ du nombre d'étoiles pour une masse donnée et $N_{tot} = 10^6$. Ainsi, j'aurai des valeurs entre 0 et 600 sur les diagrammes HR, et je devrais voir les géantes.

16 septembre 2011
J'ai changé un peu le format des données produites par le programme 'RTIGE_S_HR_cubes.py'. Il y a

maintenant pour chaque métallicité : un cube de 354 diagrammes H-R 'RTIGE_2011_HRcube_z0002.fits', un cube de paramètres stellaires 'RTIGE_2011_star_params_z0002.fits', et deux fichiers *dat donnant les types spectraux correspondants aux différentes cases du diagramme H-R (soit 100 colonnes de 100 types spectraux). Je ne parviens pas à les ajouter dans le fichier FITS des paramètres stellaires car je crois que le format FITS n'accepte pas les strings. Dans le fichier de paramètres stellaires, je donne la carte de chacun des autres paramètres stellaires donnés dans les isochrones, i.e, pour chaque case de diagramme H-R repérée par LV et Te, on a la masse, le logg, etc... Dans le cube, les images correspondent dans l'ordre à :

- logL
- logg
- mass
- R
- MU
- MB
- MV
- MR
- MI
- MJ
- MH
- MK
- rcol
- gcol
- bcol

Les unités sont les mêmes que dans les fichiers d'isochrones modifiés.

19 septembre 2011

Après avoir corrigé quelques erreurs de code, j'ai pu faire tourner le programme 'RTIGE_S_HR_cubes.py'. J'ai bien deux fichiers FITS, un contenant les diagrammes HR et un contenant les paramètres stellaires de ces étoiles. Un problème apparaît clairement pour les naines blanches : elles sont aussi lumineuses et lourdes que les géantes bleues ! Deuxième problème : des diagrammes sont entièrement vides : 1, 108, 199 pour z=0.0002 ; 1, 104, 201 pour z=0.0004. Aussi le logL des géantes rouges est aussi bas que celui des naines rouges... OK, j'ai trouvé le bug : je transposais les diagrammes HR avant de les concaténer dans le cube, alors que je ne le faisais pas pour les luminosités et températures. Du coup, le programme de récupération de paramètres stellaires prenait l'axe x pour l'axe y.
Les fichiers suivants sont donc écrits pour chaque métallicité # :

- 'RTIGE_2011_HRcube_z#.fits' : Contient les 354 diagrammes HR, images de 100x100, avec pour chaque case le nombre d'étoiles sur 1 million en log10, fois 1000.
- 'RTIGE_2011_star_params_z#.fits' : Contient les 15 données listées le 16 septembre dans le journal pour chaque case soit 15 images.
- 'RTIGE_2011_lcmap_z#.dat' : Contient
- 'RTIGE_2011_spcmap_z#.dat'

Je travaille maintenant sur l'ajout d'âges moyens et de métallicités moyennes aux particules de Fux.

20 septembre 2011

Hier soir, j'ai découvert une astuce qui simplifie tout !

On avait un soucis lorsqu'on voulait faire vieillir une étoile donnée : il fallait passer par des isochrones, les diagrammes HR ne suffisant pas. C'est très lourd.

Eh bien si en fait, ils suffisent. Avec les diagrammes HR, je donne une carte de masse, et une carte pour toutes les autres données. Pour une case donnée, on sait donc tout.

Alors, si on connait la masse de l'étoile que l'on veut faire veillir, il suffit de regarder dans l'isochrone plus vieille quelle est la case de diagramme HR non-vide dont la masse est la plus proche de la masse de l'étoile. Et le tour est joué.

Pour résumer, on s'en sort avec des données de quelques Mo au lieu d'isochrones augmentées de 300 Mo !!

21 septembre 2011

ATTENTION : pour la fonction histogram2d, pour qu'il n'y ait pas de normalisation il faut mettre norm=0 ou norm=False ! ! ! REFAIRE TOURNER TOUS LES CODES CONTENANT CETTE FONCTION UTILISANT norm='False') Les procédures de proj3D avaient la bonne écriture. Je dois refaire tourner 'RTIGE_S_HR_cubes.py'.

L'idée décrite hier pour faire vieillir les étoiles ne va pas, car une géante rouge et une naine rouge ont la même masse => dégénérescence des cas. Il faut utiliser des trajets évolutifs.

Avant de regarder ça, je mets les âges dans le modèle de Fux. J'ai trouvé la solution avec l'aide de Frédéric.
Voilà la méthode :
On a une distribution altitude,âge sous la forme (voir 19 mai 2011) :

$$f(z,a) = \alpha e^{-z/100a} \qquad a \leqslant 3\ Gyr \tag{20}$$

$$\alpha e^{-z/300} \qquad a > 3\ Gyr \tag{21}$$

(pour $z \geqslant 0$) a en Gyr, z en pc. L'intégrale de cette fonction doit faire 1 :

$$\int_{a=a_1}^{a_2}\int_{z=0}^{+\infty} f(z,a)dzda = 1 \tag{22}$$

$$\int_{a=a_1}^{3}\int_{z=0}^{+\infty} \alpha e^{-z/100a}dzda + \int_{a=3}^{a_2}\int_{z=0}^{+\infty} \alpha e^{-z/300}dzda = I + II = 1 \tag{23}$$

$$I = \int_{a=a_1}^{3} \alpha \int_{z=0}^{+\infty} e^{-z/100a}dzda = \int_{a=a_1}^{3} \alpha \left[-100ae^{-z/100a}\right]_0^{+\infty} da \tag{24}$$

$$= \int_{a=a_1}^{3} \alpha(100a)da = \left[\alpha \times 100 \times \frac{1}{2}a^2\right]_{a_1}^{3} = \alpha(300 - 50 \times a_1^2) \tag{25}$$

$$II = \int_{3}^{a_2} \alpha \left[-300e^{-z/300}\right]_0^{+inf} da = \int_{3}^{a_2} \alpha \times 300da = \left[300\alpha a\right]_3^{a_2} = 300\alpha(a_2 - 3) \tag{26}$$

$$\alpha(300 - 50a_1^2 + 300a_2 - 900) = 1 \tag{27}$$

$$\alpha = \frac{1}{300 - 50a_1^2 + 300a_2 - 900} \tag{28}$$

Pour $a_1 = 10^{6.6} \times 10^9$ et$a_1 = 10^{10.13} \times 10^9$, $\alpha \approx 0.00029$. Comme il y a très peu d'étoiles au-dessus de 4kpc (voir histogramme), on peut prendre z entre 0 et 4000pc, et garder ce α calculé entre 0 et $+\infty$.

J'ai écrit un programme ('RTIGE_S_agemap.py'), qui crée une carte d'âges dans le disque : Il contient l'algorithme suivant :
Soit 1 million d'étoiles.

```
while i<1e6 :
    z=valeur aléatoire entre 0 et 4000 pc (uniforme)
    a=valeur aléatoire entre 10^6.6 × 10^9 et 10^10.13 × 10^9 (uniforme)
    n=valeur aléatoire entre 0 et α (uniforme)
    Calcul de dn pour z et a (fonction f(z,a))
    si n<=dn :
        On garde z et a
        i+=1
```

On fait ensuite l'histogramme à 2D de z et $log10(a \times 10^9)$, 80 boîtes en ordonnées soit des boîtes de 50 pc, et 354 boîtes en abscisses, soit des boîtes de 0.01 log10(age en yr). J'obtiens pour 1 million d'étoiles la proportion pour chaque case, normalisée à 1 (voir *figure* 14). La carte d'âge est sauvée en fits : '/Modeles_galaxie/RTIGE_2011_agemap.fits'

Pour attribuer un âge aux particules du modèle de Fux il suffit ensuite de :
1) classer les particules selon leur z dans des boîtes de 50pc.
2) tirer aléatoirement selon une loi uniforme un âge entre $10^{6.6} \times 10^9$ et $10^{10.13} \times 10^9$ et un n entre 0 et le maximum de la carte d'âge.
3) si n est plus petit que la valeur de la carte d'âge pour l'âge tiré, alors on garde et on donne cet âge à la particule.

22 septembre 2011
J'écris maintenant le programme qui attribue un âge à chaque particule de Fux : 'RTIGE_S_catalogue.py' Pour les métallicités je prends : Disque : 3 métallicités différentes selon R : pour $0 < R < 3$ kpc : [Fe/H]=0.5 (Z=0.03), pour $3 < R < 8$ kpc : [Fe/H]=0 (Z=0.01), pour $R > 8$ kpc : [Fe/H]=-0.5 (Z=0.004). Halo et Bulbe : $Z = 0.0004$, $Z = 0.0008$ et $Z = 0.0002$ que je répartis aléatoirement en tirant pour chaque particule l'une de ces trois valeurs selon une loi uniforme. Le modèle de Fux est modifié et le programme écrit les fichiers 'RTIGE_2011_Fux_disc.dat' et 'RTIGE_2011_Fux_sphero.dat'. J'ai tracé l'histogramme des ages pour le disque : *figure* 15.

23 septembre 2011 :
J'ai fait une erreur d'un facteur 2 pour le calcul de L_V et de $L_{V_\odot}$!. J'ai corrigé tous les programmes. Je dois relancer 'RTIGE_S_isochrones.py' et 'RTIGE_S_HR_cubes.py'. Je les envois sur Aquilon et les lancerai lundi. C'est fait.

28 septembre 2011 :
Les données sont OK maintenant.

Pour ce qui est du vieillissement des étoiles, Misha Haywood me dit que si je prends la masse initiale et non la masse actuelle dans les isochrones, il ne peut pas y avoir d'ambiguité, car les masses sont croissantes le long d'une isochrone : il ne peut pas y avoir deux étoiles de classes de luminosité différentes ayant la même masse.

4 octobre 2011 :
J'avais fait une erreur pour la normalisation des diagrammes HR dans le code. C'est réparé. J'ai augmenté la précision des valeurs : log10(proportion)*1000. J'ai également mis les fichiers de données en ascii. J'ai modifié les programmes et les relancés. La carte âge altitude n'a pas assez de précision en nombre d'étoiles : On ne voit pas les étoiles très jeunes ($log10(age) < 7.22$). J'ai augmenté le nombre d'étoiles que le programme tire pour construire la carte : je suis passé de 10^6 à 10^8. Du coup, c'est 100 fois plus long à tourner : quelques heures au lieu de quelques minutes. La carte est construite. J'ai redistribué les âges des particules selon cette carte, et cette fois l'âge le plus petit est 6.67.

5 octobre 2011 :
Avec une carte a 2×10^8 étoiles, ça ne change pas grand chose : j'ai attribué deux fois un âge aux particules, et les deux fois l'âge minimum était différent : c'est parce que la probabilité d'avoir une étoile très jeune est très faible. Donc pour 10^6 particules, on a très peu de chances d'avoir une particule d'âge moyen en dessous de 7. Bref, pour les particules, c'est logique. La particule la plus jeune a un âge de 6.63, dans l'archive que j'envoie aujourd'hui. Quand on va éclater les particules on aura beaucoup d'étoiles, et donc on aura des étoiles jeunes.

15 novembre 2011 :
J'ai reçu les modèles basse résolution de Françoise Combes et Paola DiMatteo. J'ai une série de fichiers "sft##.dat", numérotés de 00 à 10, âgés de 0 à 500 millions d'années. Le premier modèle part d'une galaxie déjà formée. Ils contiennent des particules d'étoiles ($N_{stars} = 40000$), des particules hybrides gaz/étoiles ($N_{gas} = 40000$), et des particules de matière noire ($N_{halo} = 40000$). Les 32000 premières

particules d'étoiles correspondent au disque, les 8000 autres au bulbe et halo. Les fichiers contiennent 6 colonnes. Chaque vecteur est enregistré linéairement, chacun à la file de l'autre comme ceci :

- $x_{gas+stars+halo}$: 120000 valeurs
- $y_{gas+stars+halo}$: 120000 valeurs
- $z_{gas+stars+halo}$: 120000 valeurs
- $vx_{gas+stars+halo}$: 120000 valeurs
- $vy_{gas+stars+halo}$: 120000 valeurs
- $vz_{gas+stars+halo}$: 120000 valeurs
- $amass_{gas+stars+halo}$: 120000 valeurs
- $tnow$: 1 valeur sur la première colonne les 5 autres vides
- amg_{gas} : 40000 valeurs
- $znew_{gas}$: 40000 valeurs
- $zn_{gas+stars}$: 80000 valeurs
- $tbirth_{gas+stars}$: 80000 valeurs

J'ai écrit un programme 'RTIGE_galaxy_models_2.py' qui réécrit ces fichiers dans le format RTIGE : avec un header et des colonnes. J'ai écrit un second programme, 'RTIGE_S_catalogue.py' (l'ancien a été renommé avec la mention _old), qui ajoute les âges et modifie les métallicités des particules. Au final j'écris des fichiers 'RTIGE_2011_CombesDiMatteo_gas_stars_sft##.dat' (particules hybrides gaz/étoiles), 'RTIGE_2011_CombesDiMatteo_stars_disc_sft##.dat', 'RTIGE_2011_CombesDiMatteo_stars_bulb_sft 'RTIGE_2011_CombesDiMatteo_darkhalo_sft##.dat'.

Je travaille sur le modèle sft00. J'utilise la carte d'âges pour donner aux particules du disque un âge entre 6.6 et 9.98 (9.5 Gyr) en log10(yr). Je donne un âge de 11.5 Gyr aux particules du bulbe et 12.5 Gyr aux particules du halo. Ensuite, pour chaque modèle plus récent, j'ajoute 50 millions d'années à chaque particule.

Je regarde les métallicités (voir *figures* 16 et 17) :

Pour le disque tout est ok, mais pour le bulbe, la métallicité devrait être beaucoup plus faible. Paola me dit que c'est normal, car ils n'ont pas considéré un bulbe comme celui de la voie lactée, formé avant le disque. Je vais donc modifier les valeurs de métallicité des particules du bulbe. Je regarde la variation de la métallicité avec le rayon (voir *figure* 18) : Je vais discrétiser les valeurs de métallicité afin d'alléger le modèle. J'en choisis 5 : 0.005, 0.01, 0.015, 0.02, 0.025, 0.03. Pour chaque valeur, je regarde quelles sont les particules qui ont une métallicité à ± 0.0025, et je leur donne cette valeur. Je ne peux pas prendre les valeurs 0.035 et 0.04 car le site CMD ne permet pas de produire des isochrones à ces métallicités. La première tranche va donc de 0 à 0.0075, et la dernière de 0.0275 à 0.04.

Pour le halo et le bulbe, je prends trois valeurs : $Z = 0.0004$, $Z = 0.0008$ et $Z = 0.0002$ que je répartis aléatoirement en tirant pour chaque particule l'une de ces trois valeurs selon une loi uniforme. J'ai donc en tout 5+3=8 valeurs de métallicités. Je dois télécharger de nouvelles isochrones, et fabriquer de nouveaux cubes de paramètres stellaires. D'abord, le gaz.

Particules de gaz :

Toujours dans le programme 'RTIGE_S_catalogue.py' , je modifie les données concernant les particules hybrides gaz/étoiles. Pour ce qui est des âges, je regarde la masse d'étoiles fabriquée par tranches de 50 millions d'années. Pour chaque modèle, j'ai $M_{etoiles} = M_{tot} - M_{gaz}$. La masse totale et la masse de gaz de la particule est donnée dans le fichier. Ainsi, je connais la masse d'étoiles pour chaque particule. Je connais par soustraction des $M_{etoiles}$ d'un fichier à l'autre pour chaque particule, la masse de nouvelles étoiles pour chaque particule de chaque modèle. Ainsi je connais l'âge de toutes ces nouvelles étoiles à 50 millions d'années près. Au final, dans le modèle sft10 j'ai des étoiles de 0, 50, 100, 150, 200, 350, 400, 450 et 500 millions d'années dans les particules. Certaines qui ont commencé à former des étoiles un peu après le début de la simulation n'ont pas les plus vieilles étoiles, autrement dit, la proportion de masse d'étoiles de chaque tranche d'âge change selon la particule considérée. Dans les fichiers de modèle j'ajoute donc 11 colonnes, donnant chacune la masse d'étoiles de chaque classe

d'âge. Je donne ces classes d'âge en log10(yr), la classe d'âge 0 a l'âge le plus jeune disponible dans les fichiers d'isochrones soit 6.6. Les valeurs en log10 sont donc : 6.60, 7.70, 8.00, 8.18, 8.30, 8.40, 8.48, 8.54, 8.60, 8.65, 8.70.

Pour la métallicité, j'ai tracé zm et znew, métallicité moyenne et métallicité des nouvelles étoiles pour chaque particule en fonction de la distance au centre galactique (voir *figures* 19 et 20) . J'obtiens grosso-modo la même chose dans les deux cas. Et il semble raisonnable de prendre les mêmes métallicités que pour les particules étoiles. Je fais donc la même discrétisation.

Paramètres stellaires :
Je télécharge les isochrones pour les nouvelles métallicités. Je modifie le programme 'RTIGE_S_isochrones.py', pour qu'il travaille en plus avec les nouvelles valeurs de métallicité (on garde les anciennes, on en a besoin pour les amas ouverts). Pour les paramètres stellaires des naines blanches, j'ai moins de valeurs de métallicité différentes. J'avais associé des chemins évolutif calculés pour une certaine métallicité aux métallicités choisies pour les isochrones. Voilà comment je les associe en prenant en compte les nouvelles valeurs :
Z=numpy.array(['0002','0004','0008','003','004','005','01','015','02','0.025','03'])
Z2=numpy.array(['0001','0004','001','004','004','004','008','008','008','04','04'])
J'ai lancé le programme sur aquilon pour les 4 nouvelles valeurs de métallicité : 0.005, 0.015, 0.02 et 0.025.

16 novembre 2011 :
Les fichiers d'isochrones sont écrits, j'ai lancé le programme 'RTIGE_S_HR_cubes.py' après y avoir ajouté les nouvelles valeurs de métallicité. Les diagrammes HR sont OK, à part la présence d'étoiles autour des naines blanches ajoutées par mon algorithme. Je ne sais pas d'où viennent ces points, ils n'apparaissent que sur les nouvelles isochrones. Est-ce que quelque chose aurait changé sur leur site ? Il faudra creuser ce mystère plus tard.

Pour avoir le nombre d'étoiles par particules, il faut diviser la masse totale d'une particule par la masse moyenne donnée par la carte de masse de l'isochrone correspondant à son âge.

J'ai donc tout ce qu'il faut pour les étoiles de champ. Je m'occupe maintenant des particules hybrides gaz/étoiles. Voir document "Milieu interstellaire".

17 novembre 2011 :
J'ai modifié le programme 'RTIGE_S_catalogue.py'. J'ai ajouté une colonne "nombre moyen d'étoiles" donnant le nombre d'étoiles calculé pour une particule à partir de la masse moyenne de ses étoiles.

18 novembre 2011 :
Après correction d'erreurs de code et l'écriture d'une fonction "Nstars_mean" dans RTIGEutil.py donnant le nombre moyen d'étoiles pour une masse totale, un âge, un cube de diagramme HR et un cube de paramètres, j'ai lancé 'RTIGE_S_catalogue.py'. C'est très lent.

28 novembre 2011 :
J'ai maintenant les nombres moyens d'étoiles pour chaque particule étoile et pour chaque tranche d'âge de chaque particule hybride. J'ai voulu voir une image de galaxie en utilisant la méthode que j'ai développée avec Mikaël pour les couleurs. (voir partie méthode). Le bulbe apparaît blanc, le disque rose clair, et les particules hybrides rose foncé. Je ne parviens pas à trouver d'où ça vient. J'ai écrit un programme 'galaxy3D2.py' qui fait ça, et un programme 'test4.py' qui le fait pour 2000 particules de chaque ensemble. Pour le calcul des luminosités dans les trois canaux de couleurs, j'ai écrit deux fonctions dans 'RTIGEutil.py', une pour les particules étoiles : 'particle_colors', et une pour les particules hybrides : 'particle_colors_2'.

Je viens de corriger la gestion des cases vides des diagrammes HR : au lieu de mettre les cases à 0 pour tous les vecteurs, je ne fais les calculs que sur les cases non-vides. Les images ne sont plus roses, mais blanches. J'ai remis la multiplication par le nombre d'étoiles dans le calcul de luminosité dans les deux fonctions, et relancé le programme. On va voir. C'est toujours blanc. voir *figures* 21 (disque), 22 (bulbe), 23 (particules hybrides) et 24 (tout).

5 décembre 2011 :

Pour les âges, il faut raisonner en termes de classe d'âge, pas d'âge!!!!! les âges des particules ou amas dépendent du modèle choisi, mais pas la classe d'âge. exemple : modèle 07 : contient les classes d'âge 00 à 07, et les âges de ces classes, sont 8.54 à 6.6.

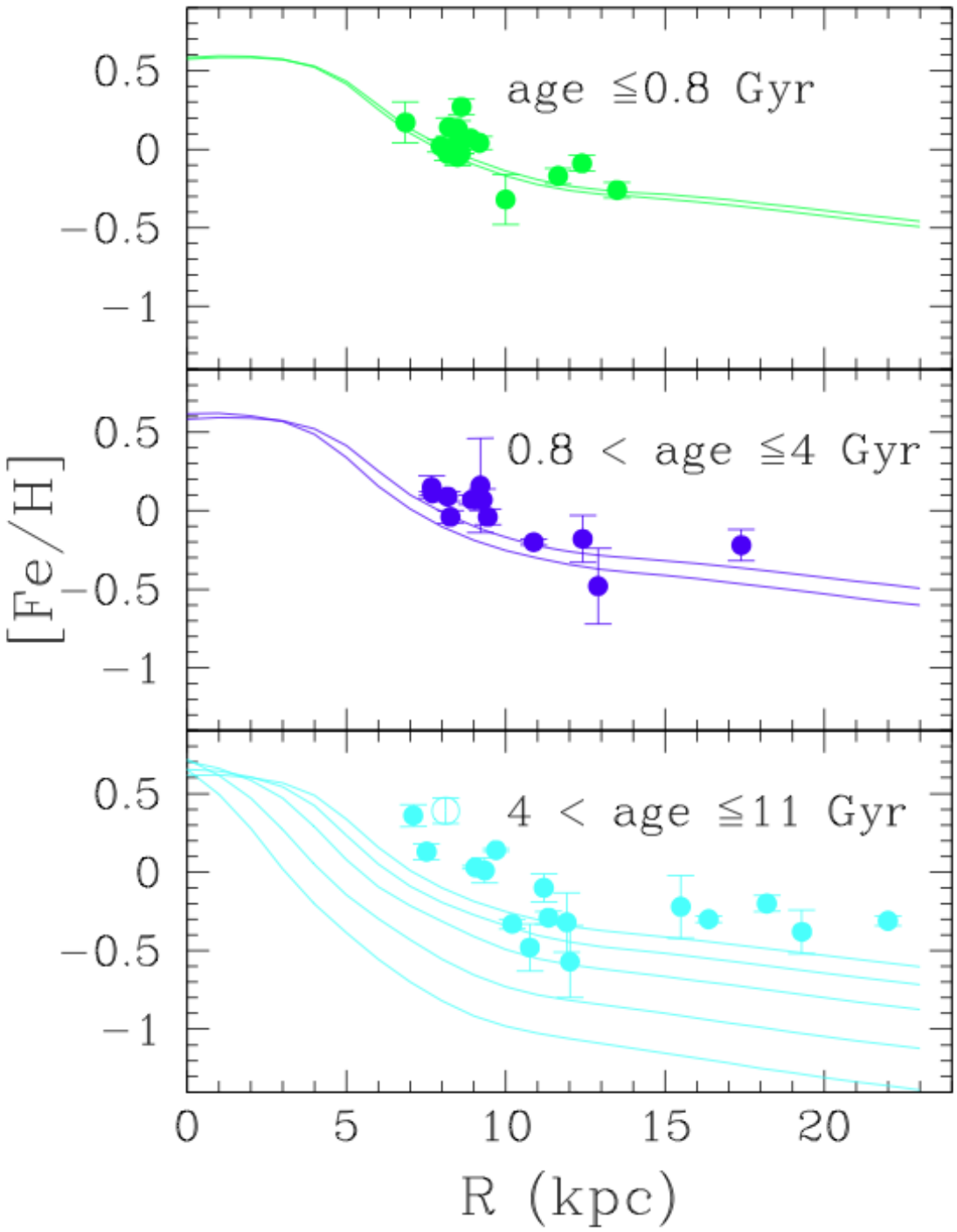


Fig. 1. Gradient of [Fe/H]: comparison between high-resolution data of OCs with the model decribed in the Appendix. Different panels show abundances for different age bins: *top* (panel 1), ages ≤0.8 Gyr; *middle* (panel 2), ages 0.8 Gyr < age ≤ 4 Gyr; *bottom* (panel 3), ages 4 Gyr < age ≤ 11 Gyr. The models adopted to compare with observations are for 0 and 1 Gyr ago (panel 1), 1 and 4 Gyr ago (panel 2), and 4, 6, 8, 10, 11 Gyr ago (panel 3).

FIGURE 10 –

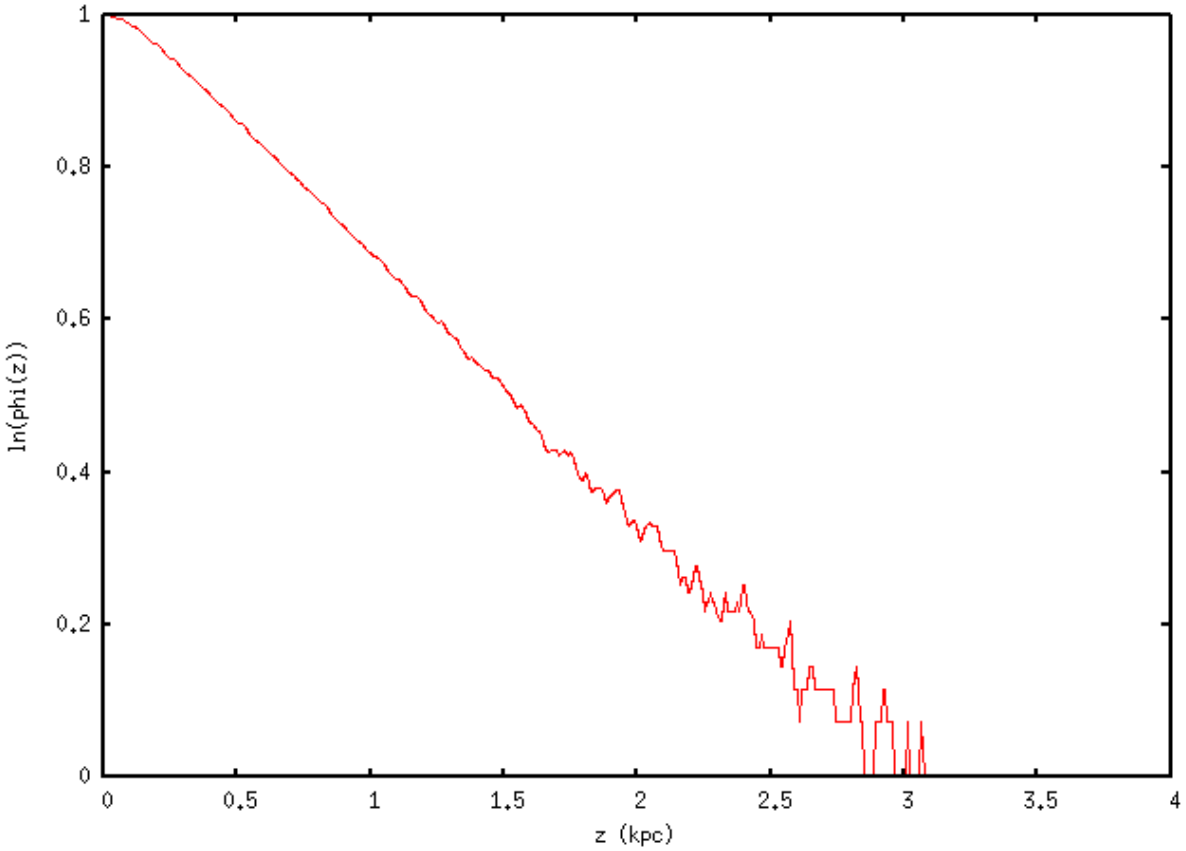


FIGURE 11 –

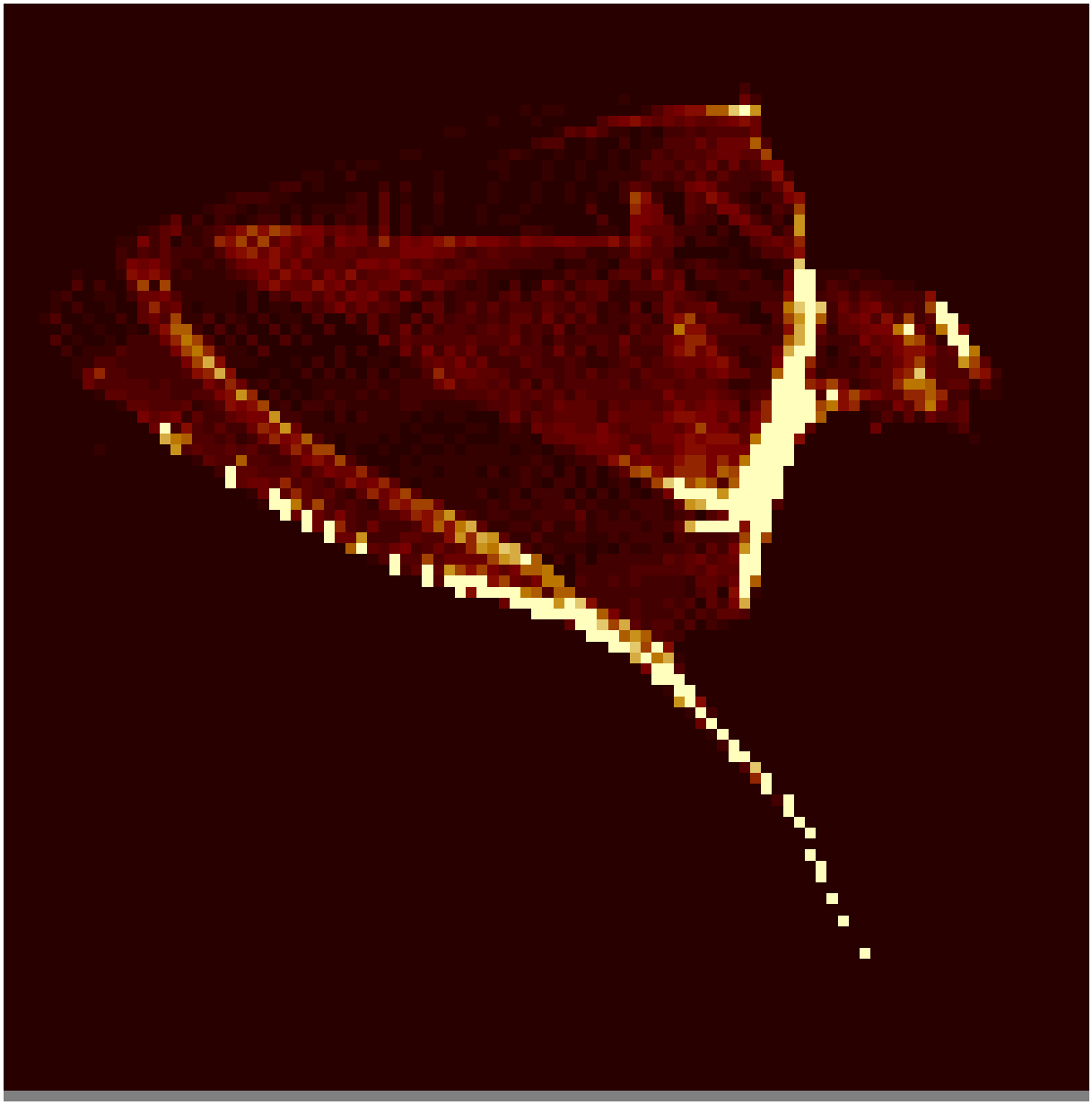


Figure 12 –

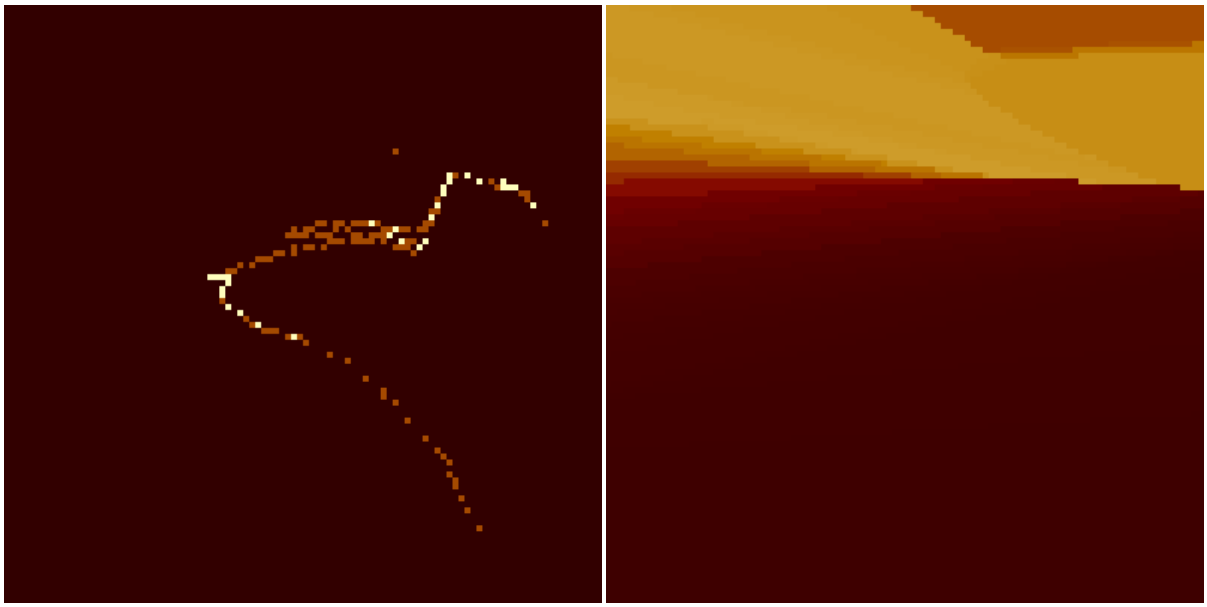


Figure 13 –



Figure 14 –

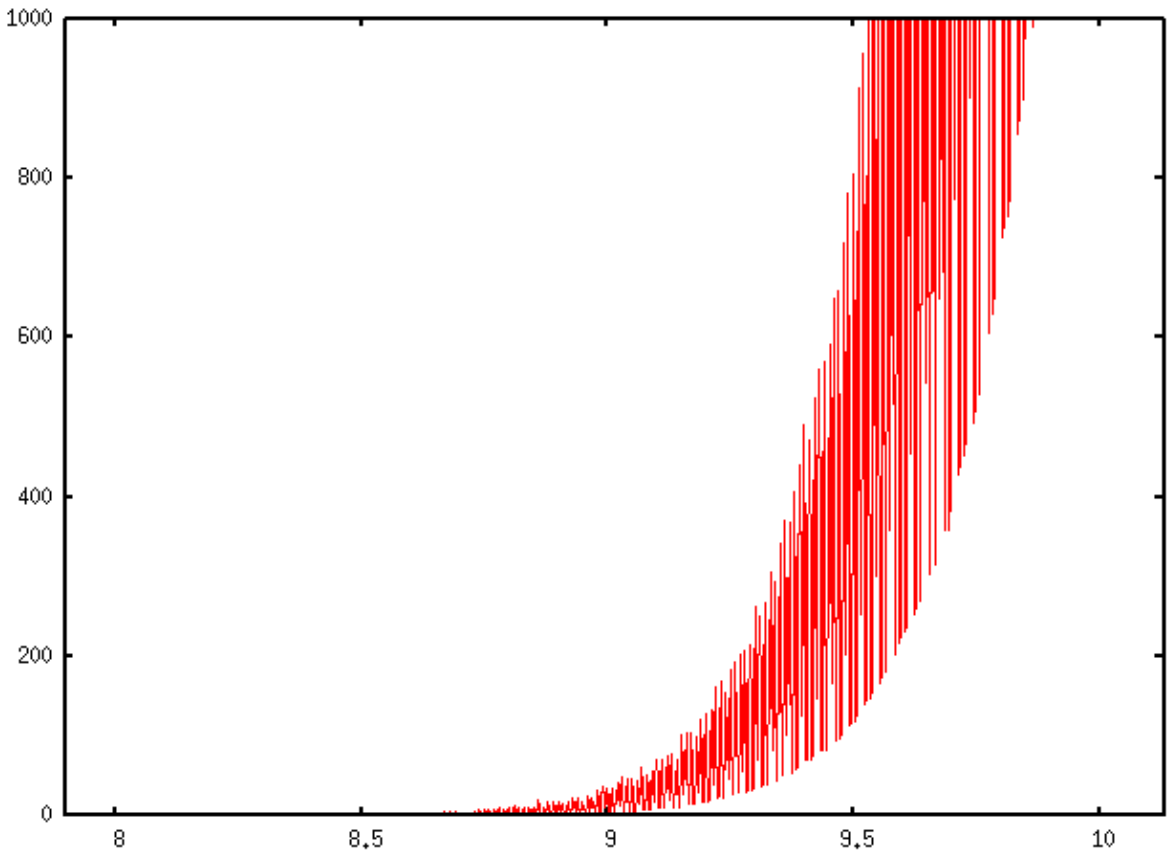


FIGURE 15 –

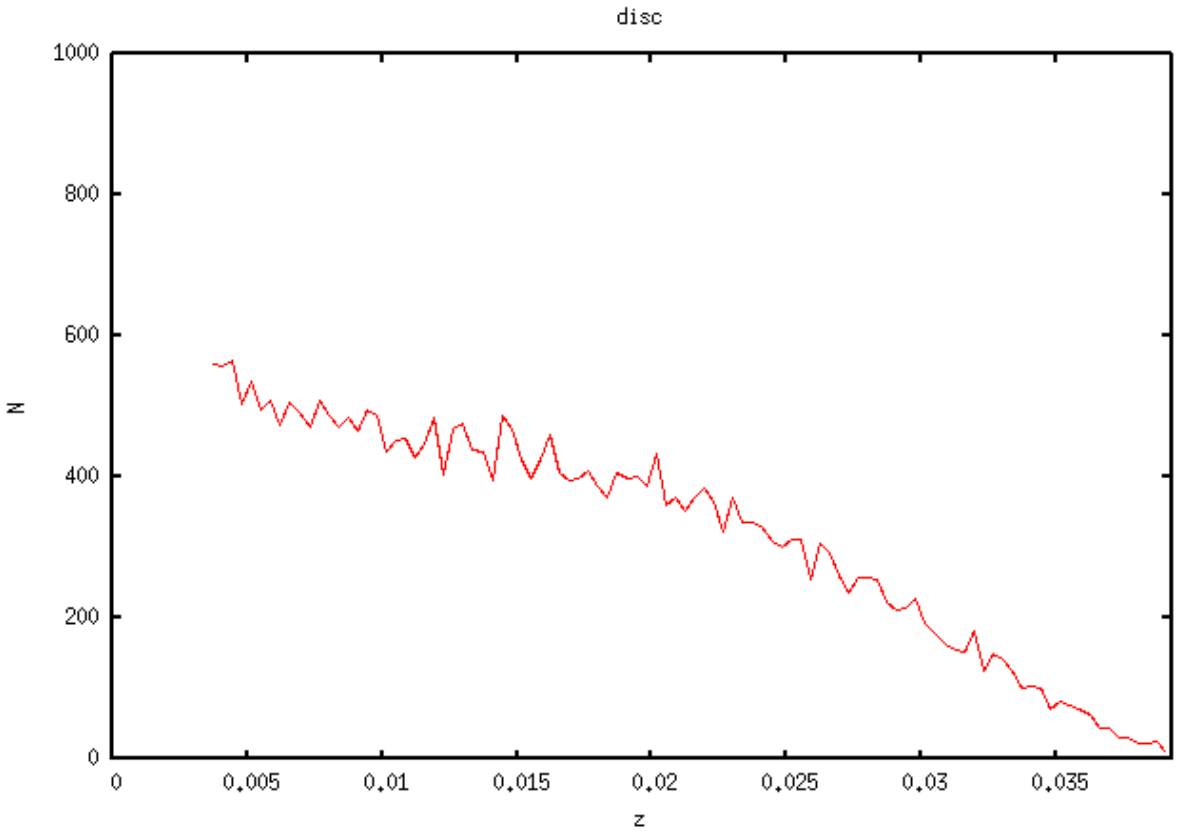


FIGURE 16 –

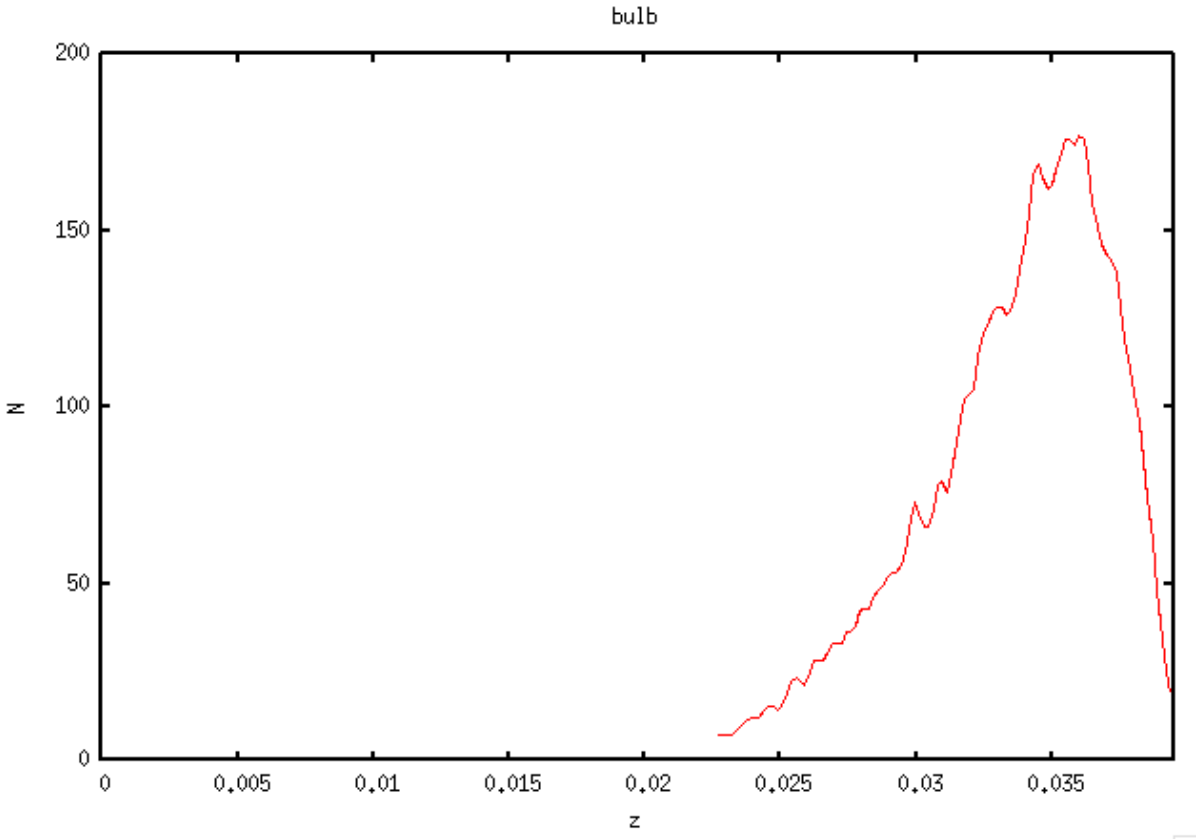


FIGURE 17 –

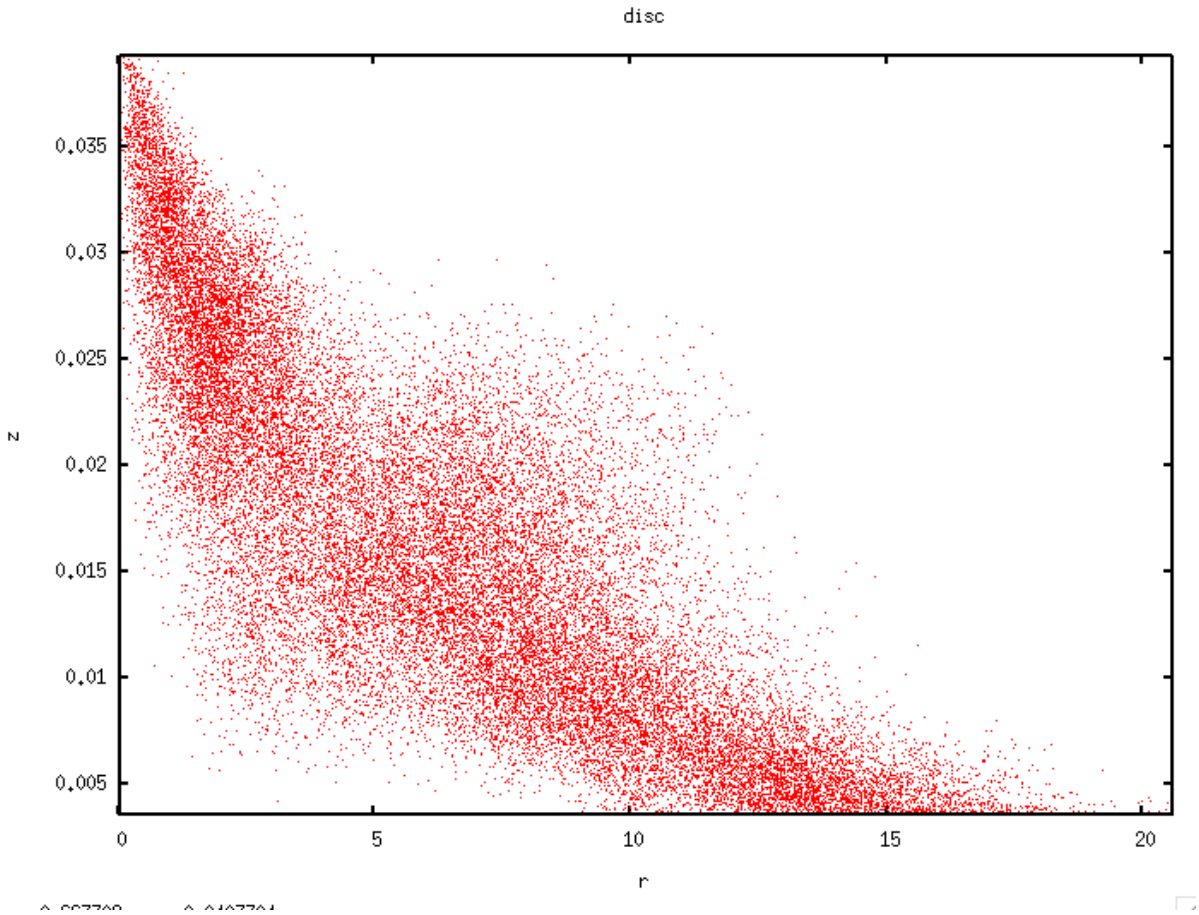


FIGURE 18 –

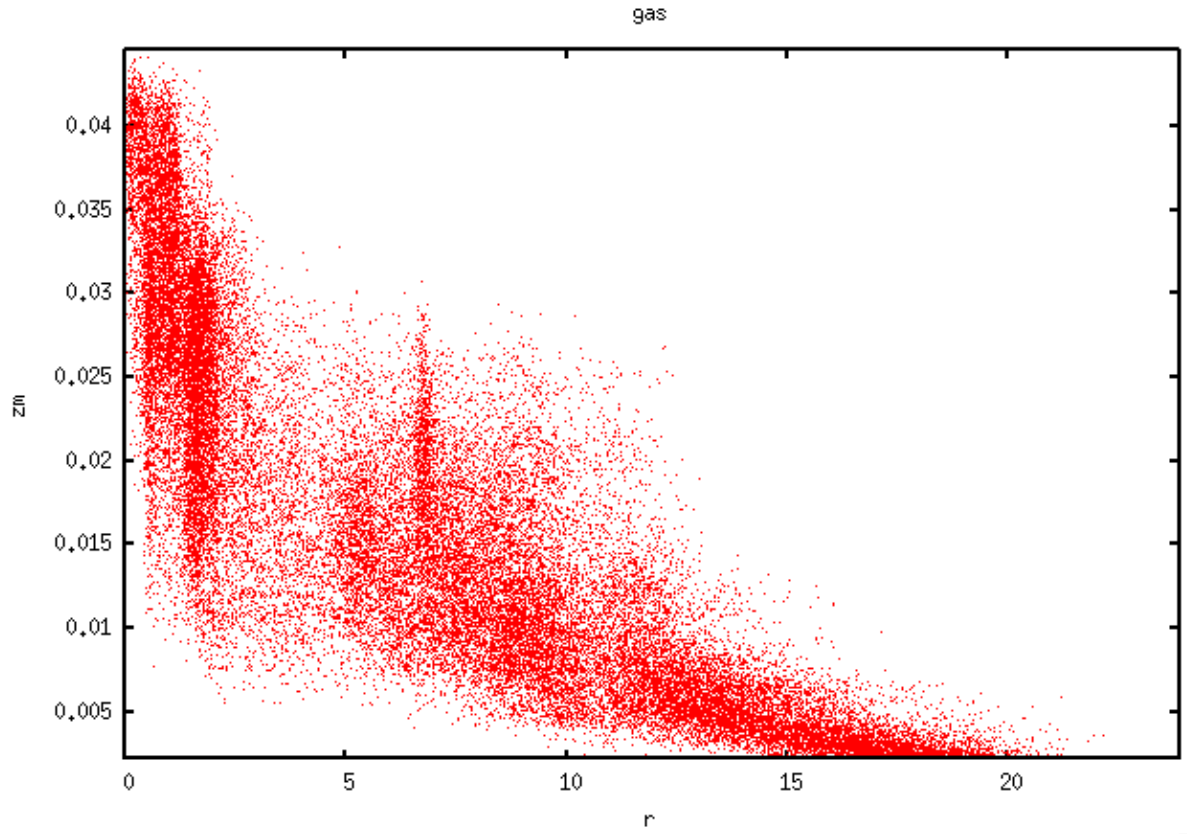


FIGURE 19 –

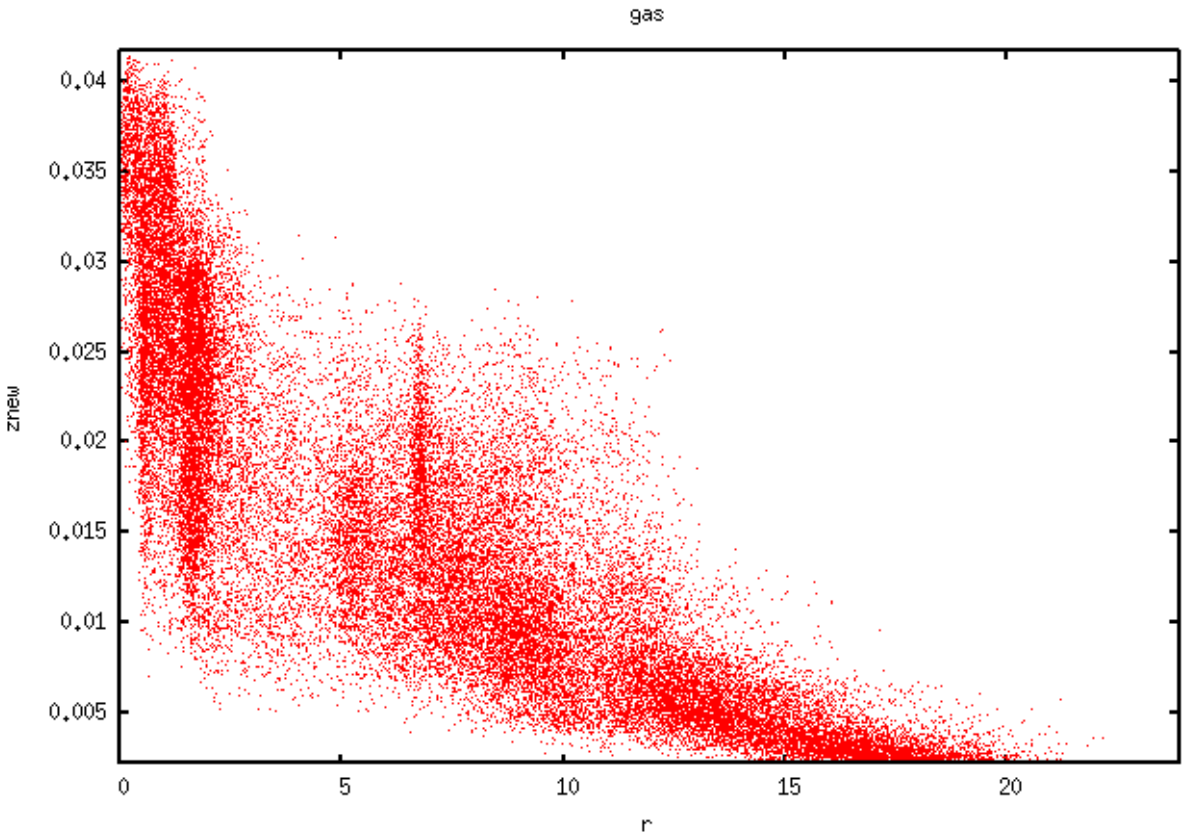


FIGURE 20 –



FIGURE 21 –

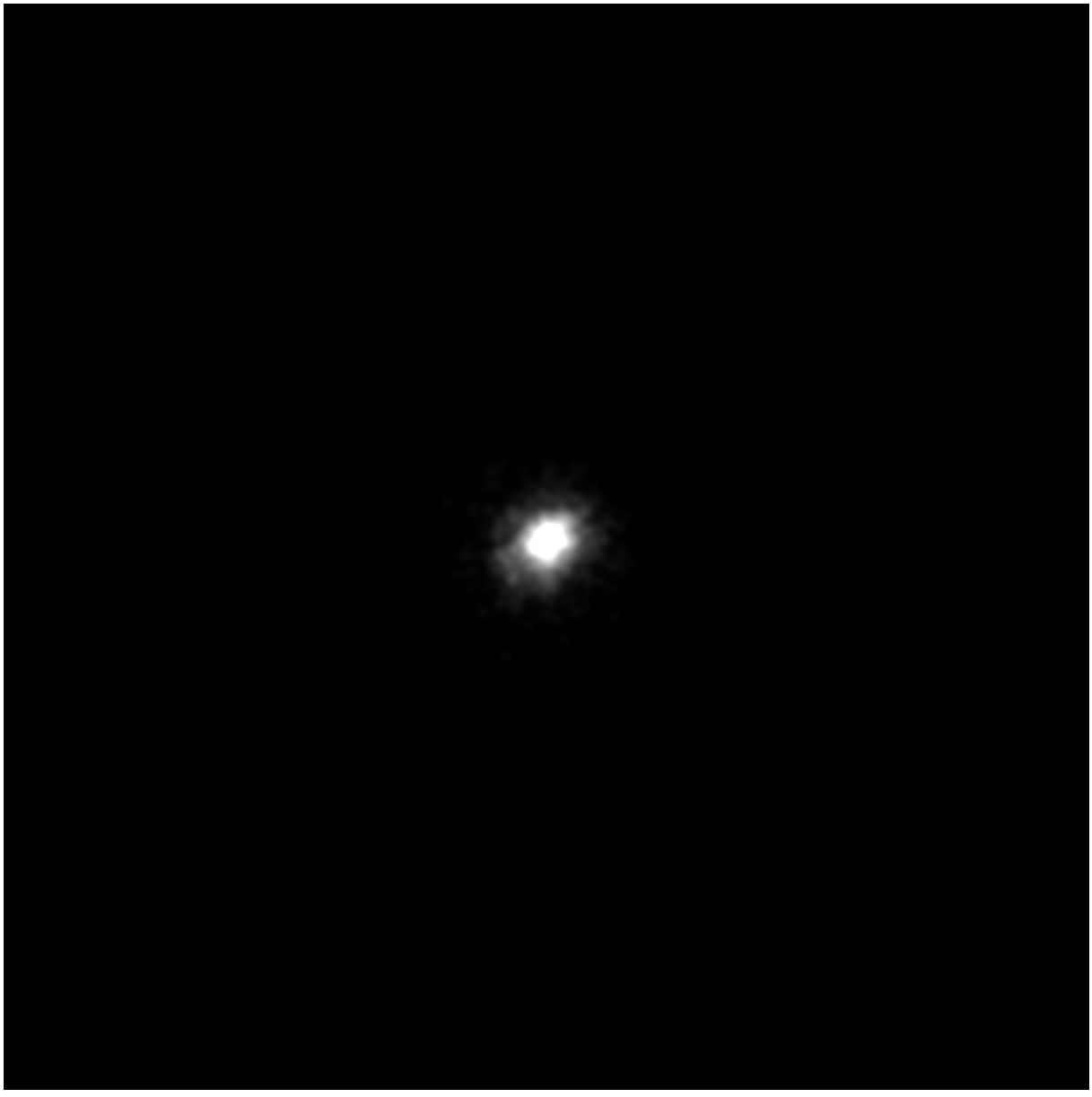


FIGURE 22 –

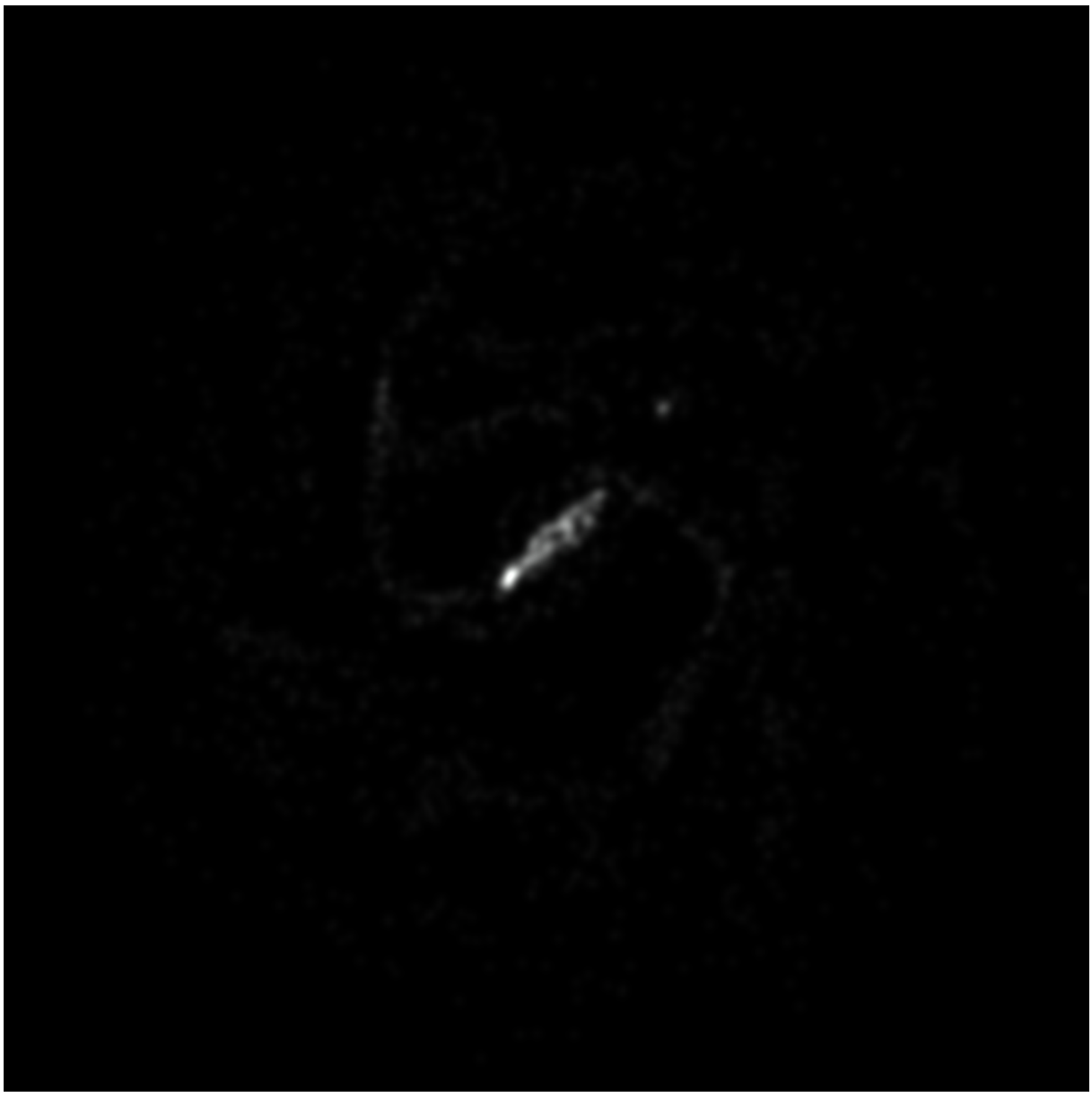


FIGURE 23 –

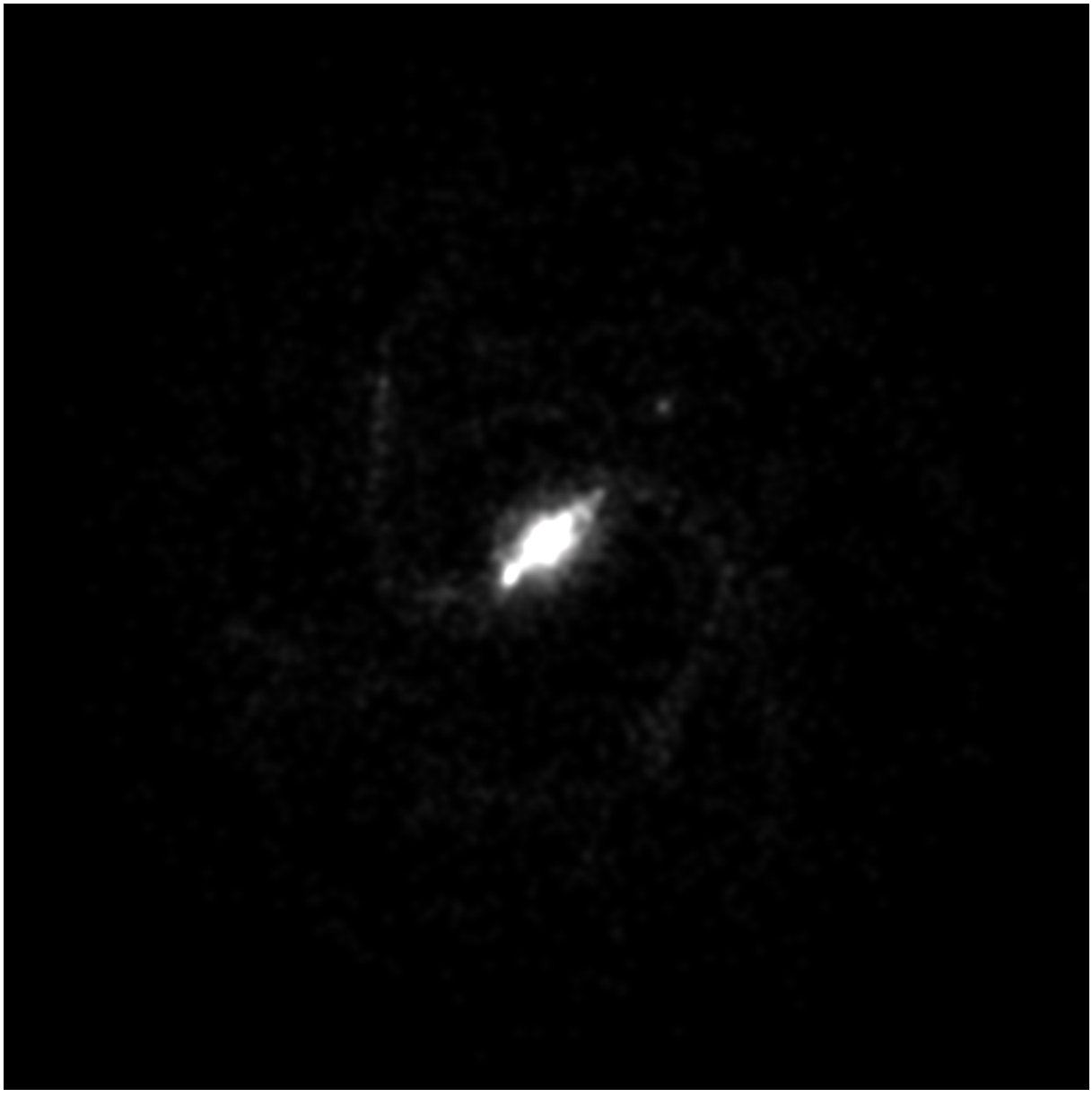


FIGURE 24 –