

# Explications des codes d'analyse

Stage été 2021

Dernière modification : 25 août 2021

---

Ce document contient les explications des codes d'analyse appliqués à des simulations de collisions de galaxies pendant l'été 2021. Certains de ces codes ont permis d'obtenir les figures présentées dans le document 'resultats\_collisions\_2021'. Les codes sont disponibles à l'adresse suivante : [https://github.com/valeriegendron/Analysis\\_codes](https://github.com/valeriegendron/Analysis_codes).

## Table des matières

<b>1</b>	<b>Codes Python</b>	<b>3</b>
1.1	age_distribution.py . . . . .	3
1.2	cm_collision.py . . . . .	3
1.3	cm_recenter_d.py . . . . .	4
1.4	cm_recenter_g.py . . . . .	4
1.5	cm_recenter_s.py . . . . .	4
1.6	collision_animation_g-s.py . . . . .	5
1.7	collision_animation_all.py . . . . .	5
1.8	combined_density_profiles.py . . . . .	6
1.9	combined_metallicity_profiles.py . . . . .	6
1.10	combined_sfr_collisions.py . . . . .	6
1.11	combined_time_density_profiles.py . . . . .	7
1.12	comparison_density_profiles.py . . . . .	7
1.13	comparison_metallicity_profiles.py . . . . .	8
1.14	comparison_velocity_distribution.py . . . . .	8
1.15	density_profiles.py . . . . .	9
1.16	distances_cms.py . . . . .	9
1.17	distances_cms_g-s.py . . . . .	9
1.18	galaxy_map_temporary.py . . . . .	9
1.19	galaxy_map_temporary-1frame.py . . . . .	10
1.20	gas_density_animation.py . . . . .	10
1.21	global_sfr_profile.py . . . . .	10
1.22	g-s-masses.py . . . . .	11
1.23	gmap_xy.py . . . . .	11
1.24	gmap_xz.py . . . . .	11
1.25	gmap_yz.py . . . . .	11

1.26	H-O-Fe-masses.py . . . . .	12
1.27	metallicity_profiles.py . . . . .	12
1.28	mean_O_Fe-Fe_H . . . . .	13
1.29	metallicity_bins_Fe_H . . . . .	13
1.30	rotation_curves_comparison.py . . . . .	14
1.31	separation_for_comparison.py . . . . .	14
1.32	sfr_collisions.py . . . . .	14
1.33	sfr_profile.py . . . . .	15
1.34	smap.py . . . . .	15
1.35	time_metallicity_profiles.py . . . . .	15
1.36	velocity_dispersion.py . . . . .	16
1.37	velocity_dispersion_animation.py . . . . .	16
1.38	velocity_disp_and_evol_animation.py . . . . .	16
1.39	velocity_distribution_comparison.py . . . . .	17
1.40	warp.py . . . . .	17
<b>2</b>	<b>Codes Fortran</b>	<b>18</b>
2.1	detilt.f . . . . .	18
2.2	sfr_profile.f . . . . .	18
2.3	sigma_Vg.f . . . . .	18
2.4	sigma_Vs.f . . . . .	19
2.5	split.f . . . . .	19
2.6	warp.f . . . . .	20
2.7	Z_fibre.f . . . . .	20

# 1 Codes Python

## 1.1 age\_distribution.py

*Auteur : Valérie Gendron*

Permet d'obtenir la distribution d'âges des particules d'étoiles de la galaxie impacteur à un temps donné et à une distance donnée du centre de masse du système. On choisit un temps où on voit clairement deux groupes distincts d'étoiles de l'impacteur à l'aide de l'animation produite par le code 'collision\_animation\_g-s.py'.

**Paramètres à modifier au besoin :**

- fileName : nom du fichier contenant les particules d'étoiles de l'impacteur au temps choisi (par exemple, fileName = 's000062\_2').
  - galaxy\_name : nom de la simulation choisie (par exemple, galaxy\_name = 'Af\_v1').
  - lim : distance en  $x$  (en kpc) du centre de masse du système qui délimite les deux régions d'intérêt. Par exemple, si lim = 10, on obtiendra la distribution d'âge des étoiles se trouvant à  $-10 \leq x \leq 10$  kpc et la distribution d'âge des étoiles se trouvant à  $x \leq -10$  kpc et à  $x \geq 10$  kpc du centre de masse du système.
  - bins\_center et bins\_impactor : nombre de bandes (bins) contenus dans l'histogramme pour les deux distributions, celle du centre et celle à l'extérieur, respectivement.
  - w : largeur des bandes de l'histogramme (par exemple, w = 0.005).
  - time : temps choisi de la simulation en Gyr (par exemple, time = 0.3).
  - fileNames : liste contenant les noms des fichiers 'd', 'g' et 's' correspondant au temps choisi. Cette liste se trouve dans la section "Center of mass of system". Par exemple, fileNames = ['d000062', 'g000062', 's000062'].
- \* Modifie les unités de l'âge des particules d'étoiles afin qu'elle soit en Gyr ( $10^9$  ans) et non en  $4.71 \times 10^8$  ans.

## 1.2 cm\_collision.py

*Auteur : Valérie Gendron*

Calcule et porte en graphique la position et la vitesse du centre de masse du système (parcourt l'ensemble des fichiers de données brutes contenus dans le dossier de la simulation précisée). La position et la vitesse du centre de masse de l'ensemble des particules est illustré ainsi que la position et la vitesse des centres de masses spécifiques (des particules de gaz seulement, des particules d'étoiles seulement et des particules de matière sombre seulement). \*Assume que les unités des fichiers brutes sont en 100 kpc pour la position et en 207.4 km/s pour la vitesse et les convertit respectivement en kpc et en km/s avant de les porter en graphique.

**Paramètres à modifier au besoin :**

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'Af\_v1'). Doit correspondre au nom du dossier contenant les fichiers de données brutes 'd', 'g' et 's' de la simulation.
  - for fileName in os.listdir(str(galaxy\_name)) : s'assurer que 'str(galaxy\_name)' corresponde bien au dossier contenant les fichiers de données brutes.
- \* Les légendes des graphiques devraient être améliorées.

### 1.3 cm\_recenter\_d.py

*Auteur : Valérie Gendron*

Pour les particules de matière sombre : corrige les positions (x, y, z) pour que la position du centre de masse du système soit toujours (0, 0, 0) et corrige les vitesses (vx, vy, vz) afin que la vitesse du centre de masse soit toujours nulle. Parcourt l'ensemble des fichiers de données brutes 'd' et écrit de nouveaux fichiers 'd' en les plaçant dans le dossier 'data\_centered' (doit être créé manuellement avant d'exécuter le code).

**Paramètres à modifier au besoin :**

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'Af\_v1'). Doit correspondre au nom du dossier contenant les fichiers de données brutes 'd' de la simulation.
- for fileName in os.listdir(str(galaxy\_name)) : s'assurer que 'str(galaxy\_name)' correspond bien au dossier contenant les fichiers de données brutes.
- datafile\_path : doit contenir le path (à partir de l'emplacement du fichier 'cm\_recenter\_d.py') des nouveaux fichiers 'd' qui seront obtenus. Par exemple, datafile\_path = str(galaxy\_name) + "/data\_centered/" + str(fileName), ce qui pourrait donner I\_hh/data\_centered/d000000.

\* Ne modifie pas les unités. Les positions sont encore en unités de 100 kpc et les vitesses en unités de 207.4 km/s.

\* Attention : si l'on souhaite recentrer l'ensemble des particules d'une simulation avec collision, il est préférable d'utiliser le code Fortran 'detilt.f', qui recentre toutes les particules sur le centre de masse des étoiles, plutôt que d'exécuter les codes 'cm\_recenter\_d.py', 'cm\_recenter\_g.py' et 'cm\_recenter\_s.py', qui recentrent les particules de matière sombre, de gaz et d'étoiles sur leur centre de masse respectif. Ces codes peuvent cependant être utilisés pour les simulations isolées, puisque la position des centres de masse respectifs des particules de matière sombre, de gaz et d'étoiles est sensiblement la même.

### 1.4 cm\_recenter\_g.py

*Auteur : Valérie Gendron*

Même chose que 'cm\_recenter\_d.py', mais pour les particules de gaz au lieu des particules de matière sombre.

### 1.5 cm\_recenter\_s.py

*Auteur : Valérie Gendron*

Même chose que 'cm\_recenter\_d.py', mais pour les particules d'étoiles au lieu des particules de matière sombre.

## 1.6 collision\_animation\_g-s.py

*Auteur : Valérie Gendron*

Produit une animation des particules de gaz et d'étoiles de la simulation précisée et permet donc d'observer son évolution dans le temps. Les points bleu foncé, bleu pâle, jaunes et rouges représentent respectivement les particules d'étoiles de l'impacteur, les particules de gaz de l'impacteur, les particules d'étoiles de la galaxie cible et les particules de gaz de la galaxie cible.

### À faire avant de pouvoir exécuter le code (sur Stargate) :

- Créer et activer un environnement virtuel dans le dossier où 'collision\_animation\_all.py' se trouve. Y installer 'numpy', 'matplotlib' et 'ffmpeg' avec pip install. (Commandes : (1) python3 -m venv env; (2) source env/bin/activate; (3) python3 -m pip install numpy; (4) python3 -m pip install matplotlib; (5) python3 -m pip install ffmpeg.)  
<https://packaging.python.org/guides/installing-using-pip-and-virtual-environments/?fbclid=IwAR3kJLqW-D0r>
- Si droits d'administration sur Stargate, installer ffmpeg en utilisant la commande 'sudo apt install ffmpeg'. Sinon, installer un 'static build' de ffmpeg (<https://johnvansickle.com/ffmpeg/>) sur son propre ordinateur, puis transférer le dossier sur Stargate avec FileZilla (par exemple). (J'ai déjà installé ffmpeg dans /home/vagen16/. Vous pouvez donc simplement copier le dossier dans la directory désirée ou bien préciser le path (voir prochaine étape)).
- Dans le fichier python du code ('collision\_animation\_all.py'), préciser le path du fichier ffmpeg *exécutable*. Par exemple, `mpl.rcParams['animation.ffmpeg.path'] = r'/home/vagen16/ffmpeg/ffmpeg'`. Ce path fonctionne peu importe l'emplacement de 'collision\_animation\_g-s.py' sur Stargate.
- S'assurer que l'environnement virtuel créé plus tôt est activé avant d'exécuter le code.

### Paramètres à modifier au besoin :

- `mpl.rcParams['lines.markersize']` : contrôle la taille des particules illustrées dans l'animation.  
Par exemple, `mpl.rcParams['lines.markersize'] = 0.2`.
- `galaxy_name` : assigner le nom de la simulation désirée (par exemple, `galaxy_name = 'Af_v1'`). Doit correspondre au nom du dossier contenant les fichiers de données brutes 'g' et 's' de la simulation.
- `plan` : choisir entre 'xy', 'xz' ou 'yz'. Détermine quel plan sera animé.
- `lim` : limite des axes de l'animation en kpc. Par exemple, `lim = 50`; l'axe 'x' ira de -50 à 50 kpc et l'axe 'y' ira de -50 à 50 kpc.
- `t` : contrôle la transparence des particules illustrées dans l'animation. Par exemple, `t = 0.2`.
- `fps` : Nombre de frames par seconde (contrôle la vitesse de l'animation). Se trouve à l'avant dernière ligne du code dans l'assignation de 'writervideo'. Par exemple, `writervideo = animation.FFMpegWriter(fps=5)`.

\* Modifie les unités de la position des particules afin qu'elle soit en kpc et non en 100 kpc.

\* Prend en compte le fait que le fichier 's000000' est vide.

\*\* Une légende devrait être ajoutée.

## 1.7 collision\_animation\_all.py

*Auteur : Valérie Gendron*

Fonctionne comme le code de 'collision\_animation\_g-s.py' mais anime aussi les particules de matière sombre du système en plus des particules de gaz et d'étoiles.

## 1.8 combined\_density\_profiles.py

*Auteur : Valérie Gendron*

(Voir 'density\_profiles.py'). Trace sur un même graphique le profil de densité de multiples simulations.

### Paramètres à modifier au besoin :

- content : choisir 'g' ou 's' pour obtenir le graphique des densités de gaz ou d'étoiles, respectivement.
- galaxy\_names : liste contenant le nom de toutes les simulations désirées (par exemple, galaxy\_names = ['Af\_v1', 'Af\_v2', 'Af\_v3']). Doit correspondre au nom du dossier contenant les fichiers de données 'g' et 's'.

## 1.9 combined\_metallicity\_profiles.py

*Auteur : Valérie Gendron*

(Voir 'metallicity\_profiles.py'). Trace sur un même graphique la métallicité de la simulation désirée selon la distance au centre de masse pour différents temps.

### Paramètres à modifier au besoin :

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'Af\_v1'). Doit correspondre au nom du dossier contenant les fichiers de données 'g' et 's'.
- time\_list : liste contenant les temps auxquels la métallicité sera tracée (par exemple, time\_list = [0.3, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0]).
- dlim : limite en kpc du disque selon l'axe z (par exemple, dlim = 5). Le code ne tient compte que des particules ayant un z entre -dlim et +dlim.
- rlim : plus grand rayon [kpc] d'anneau considéré dans le plan xy du disque (par exemple, rlim = 20).
- content : choisir 'g' ou 's' pour obtenir le graphique de la métallicité du gaz ou des étoiles, respectivement.
- metallicity : choisir 'O\_H', 'Fe\_H' ou 'O\_Fe' pour obtenir respectivement [O/H], [Fe/H] ou [O/Fe] en fonction de la distance au centre de masse du système.

\* Le code a été écrit en assumant que les fichiers 'g' et 's' utilisés ont été préalablement modifiés avec 'detilt.f' afin que le moment cinétique soit en z seulement, que la position et la vitesse du centre de masse soient nulles, que les positions soient en unités de kpc et que les masses soient en unités de masses solaires.

\*\* Utilise le fichier 'tzstep.dat'.

## 1.10 combined\_sfr\_collisions.py

*Auteur : Valérie Gendron*

(Voir 'sfr\_collisions.py'). Montre sur un même graphique le taux de formation stellaire selon le temps pour l'ensemble du système des simulations précisées.

### Paramètres à modifier au besoin :

- galaxy\_names : assigner le nom des simulations désirées (par exemple, galaxy\_names = ['Af\_v1', 'Af\_v2', 'Af\_v3']). Doit aussi correspondre aux noms des dossiers contenant les fichiers de données brutes 's' des simulations, ainsi que les fichiers 'tzstep.dat'.

\* Modifie les unités de masse et de temps afin d'afficher le taux de formation stellaire en unités de masses solaires par an.

## 1.11 combined\_time\_density\_profiles.py

*Auteur : Valérie Gendron*

(Voir 'comparaison\_density\_profiles.py'). Trace sur un même graphique les profils de densité d'une simulation précisée à plusieurs temps. Fonctionne de la même manière que le code 'comparaison\_density\_profiles.py'.

### Paramètres à modifier au besoin :

- content : choisir 'g' ou 's' pour obtenir le graphique des densités de gaz ou d'étoiles, respectivement.
- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'Af\_v1').
- time\_list : liste contenant les temps en Ga auxquels les profils de densité seront obtenus (par exemple, time = [0.3, 0.48, 1.0, 2.0, 3.0]).

\* Pour les simulations avec collision, si on choisit un temps où l'impacteur n'est pas fusionné à la galaxie cible, il peut être nécessaire de d'abord séparer les particules entre les galaxies avant d'obtenir les profils de densité. Voir 'separation\_for\_comparaison.py'.

\*\* Le code a été écrit en assumant que les fichiers 'g' et 's' utilisés ont été préalablement modifiés avec 'detilt.f' afin que le moment cinétique soit en z seulement, que la position et la vitesse du centre de masse soient nulles, que les positions soient en unités de kpc et que les masses soient en unités de masses solaires.

\*\*\* Utilise le fichier 'tzstep.dat' de la simulation précisée.

## 1.12 comparaison\_density\_profiles.py

*Auteur : Valérie Gendron*

Trace sur un même graphique les profils de densité de plusieurs simulations à un temps précisé. Essentiellement la même chose que "combined\_density\_profiles.py", mais est compatible avec les simulations où la galaxie cible (ou l'impacteur) est isolée.

### Paramètres à modifier au besoin :

- content : choisir 'g' ou 's' pour obtenir le graphique des densités de gaz ou d'étoiles, respectivement.
- time : temps en Ga auquel on veut obtenir les profils de densité.
- galaxy\_names : liste contenant le nom de toutes les simulations désirées (par exemple, galaxy\_name = ['I', 'Af\_v1', 'Af\_v2']). Doit correspondre au nom du dossier contenant les fichiers de données 'g' et 's'.

\* Pour les simulations avec collision, si on choisit un temps où l'impacteur n'est pas fusionné à la galaxie cible, il peut être nécessaire de d'abord séparer les particules entre les galaxies avant d'obtenir des profils de densité. Voir 'separation\_for\_comparaison.py'.

\*\* Le code a été écrit en assumant que les fichiers 'g' et 's' utilisés ont été préalablement modifiés avec 'detilt.f' afin que le moment cinétique soit en z seulement, que la position et la vitesse du centre de masse soient nulles, que les positions soient en unités de kpc et que les masses soient en unités de masses solaires.

\*\*\* Utilise les fichiers 'tzstep.dat' de chaque simulation précisée.

### 1.13 comparison\_metallicity\_profiles.py

*Auteur : Valérie Gendron*

Trace les profils de métallicité de plusieurs simulations (incluant les simulations isolées) à un temps précisé. Une première figure est produite pour  $[O/H]$  selon la distance au centre de masse, une deuxième pour  $[Fe/H]$  et une troisième pour  $[O/Fe]$ .

#### Paramètres à modifier :

- content : choisir 'g' ou 's' pour obtenir les profils de métallicité du gaz ou des étoiles, respectivement.
- time : temps en Ga auquel on veut obtenir les profils de métallicité.
- galaxy\_names : liste contenant le nom de toutes les simulations désirées (par exemple, galaxy\_name = ['I', 'Af\_v1', 'Af\_v2']). Doit correspondre au nom du dossier contenant les fichiers de données 'g' et 's'.

\* Pour les simulations avec collision, si on choisit un temps où l'impacteur n'est pas fusionné à la galaxie cible, il peut être nécessaire de d'abord séparer les particules entre les galaxies avant d'obtenir des profils de métallicité. Voir 'separation\_for\_comparison.py'.

\*\* Le code a été écrit en assumant que les fichiers 'g' et 's' utilisés ont été préalablement modifiés avec 'detilt.f' afin que le moment cinétique soit en z seulement, que la position et la vitesse du centre de masse soient nulles, que les positions soient en unités de kpc et que les masses soient en unités de masses solaires.

\*\*\* Utilise les fichiers 'tzstep.dat' de chaque simulation précisée.

### 1.14 comparison\_velocity\_distribution.py

*Auteur : Valérie Gendron*

Trace sur une même figure les dispersions des vitesses en coordonnées cylindriques selon la distance au centre de masse du système pour les particules de gaz ou d'étoiles de plusieurs simulations à un temps précisé. Essentiellement la même chose que la fonction 'velocity\_dispersion' du code Python 'velocity\_dispersion.py', mais combine les dispersions de vitesses de plusieurs simulations. Le code produit autant de figures que de temps précisés.

#### Paramètres à modifier au besoin :

- galaxy\_names : liste contenant le nom de toutes les simulations qu'on souhaite comparer (par exemple, galaxy\_name = ['I', 'Af\_v1', 'Af\_v2']).
- time\_list : liste contenant les temps en Ga auxquels on veut obtenir des figures (par exemple, time\_list = [0.48, 1.0, 2.0, 3.0]). Les temps choisis doivent se retrouver dans le fichier d'output 'sigma\_vgc.out' ou 'sigma\_vsc.out' de chacune des simulations figurant dans la liste galaxy\_names.
- marker\_styles : liste contenant les styles de marqueurs utilisés pour chaque simulation précisée. La simulation dont le nom est à l'index 0 de la liste galaxy\_names sera représentée sur les figures par le marqueur à l'index 0 de la liste marker\_styles (et ainsi de suite). Par exemple : marker\_styles = ["o", "X", "d"].
- content : choisir 'g' ou 's' pour obtenir les dispersions de vitesses des particules de gaz ou d'étoiles, respectivement.
- fileName : (se trouve à l'intérieur de la fonction 'velocity\_dispersion') doit correspondre au nom de la directory du fichier d'output 'sigma\_vgc.out' ou 'sigma\_vsc.out' à utiliser pour chacune des simulations.

\* Pour obtenir les fichiers d'output 'sigma\_vgc.out' et 'sigma\_vsc.out', utiliser les codes Fortran 'sigma\_Vg.f' et 'sigma\_Vs.f' respectivement.



## 1.15 density\_profiles.py

*Auteur : Valérie Gendron*

Montre la densité en  $M_{\odot}/\text{kpc}^2$  de gaz ou d'étoiles au temps final de la simulation choisie. La densité est calculée dans des anneaux concentriques à partir du centre de masse du système (anneaux de 0.5 kpc d'épaisseur, de 0 à 20 kpc).

### Paramètres à modifier au besoin :

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'Af\_v1'). Doit correspondre au nom du dossier contenant les fichiers de données 'g' et 's'.
- dlim : limite en kpc du disque selon l'axe z (par exemple, dlim = 5). Le code ne tient compte que des particules ayant un z entre -dlim et +dlim.
- rlim : plus grand rayon [kpc] d'anneau considéré dans le plan xy du disque (par exemple, rlim = 20).
- content : choisir 'g' ou 's' pour obtenir le graphique des densités de gaz ou d'étoiles, respectivement.
- fileName : (situé dans la première if clause et dans la première elif clause) directory du fichier 'g' et 's' correspondant au dernier temps de la simulation (par exemple, fileName = str(galaxy\_name) + "/g000564r" et fileName = str(galaxy\_name) + "/s000564r", ce qui pourrait par exemple donner Af\_v1/g000564r et Af\_v1/s000564r).

\* Le code a été écrit en assumant que les fichiers 'g' et 's' utilisés ont été préalablement modifiés avec 'detilt.f' afin que le moment cinétique soit en z seulement, que la position et la vitesse du centre de masse soient nulles, que les positions soient en unités de kpc et que les masses soient en unités de masses solaires.

## 1.16 distances\_cms.py

*Auteur : Valérie Gendron*

Illustre la différence de distance (en kpc) entre les centres de masse des galaxies cibles et impacteur selon le temps. Sert à déterminer à quels temps de la simulation la galaxie impacteur traverse la galaxie cible.

### Paramètres à modifier au besoin

- galaxy\_name : assigner le nom de la simulation dériée (par exemple, galaxy\_name = 'Af\_v1'). Doit correspondre au nom du dossier contenant les fichiers de données brutes d', 'g' et 's' de la simulation, ainsi que le fichier 'tzstep.dat'.
- for fileName in os.listdir(str(galaxy\_name)) : s'assurer que 'str(galaxy\_name)' correspond bien au dossier contenant les fichiers de données brutes.

\* Modifie les unités de position afin d'afficher la distance entre les centres de masse en kpc et non en unités de 100 kpc.

## 1.17 distances\_cms\_g-s.py

*Auteur Valérie Gendron*

Même chose que 'distances\_cms.py', mais n'utilise que les fichiers de données 'g' et 's' pour calculer le centre de masse (au lieu d'utiliser les fichiers 'd', 'g' et 's').

## 1.18 galaxy\_map\_temporary.py

*Auteur : Valérie Gendron*

Illustre, par des nuages de points, le système précisé à 6 temps différents. Présentement (16 juin 2021) seulement utilisé pour se donner une idée générale de l'évolution du système. Pour obtenir de belles figures, utiliser plutôt le code 'galaxy\_map\_temporary-1frame.py'.

\* Ce code n'a pas été écrit pour être exécutable sur Stargate puisqu'il est seulement utilisé pour visualiser rapidement une simulation à différents temps (le code utilise 'matplotlib.pyplot.show()' et ne sauvegarde pas la figure).

## 1.19 galaxy\_map\_temporary-1frame.py

*Auteur : Valérie Gendron*

Illustre, par des nuages de points, le système précisé à un temps précis. Fonctionne présentement (16 juin 2021) seulement avec les simulations où il y a collision de galaxies (une galaxie cible et un impacteur).

### Paramètres à modifier au besoin :

- columns : (0, 1) pour plan xy, (0, 2) pour plan xz et (1, 2) pour plan yz.
- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'Af\_v1').
- time : temps en Ga auquel on souhaite voir la simulation (par exemple, time = 1.28).
- lim : limite des axes de la figure. Par exemple, lim = 50; l'axe 'x' ira de -50 à 50 kpc et l'axe 'y' ira de -50 à 50 kpc.
- Les noms des fichiers de données doivent présentement (16 juin 2021) être modifiés manuellement dépendamment du temps souhaitant être illustré.
- marker\_size : contrôle la taille des particules illustrées sur la figure. Par exemple, marker\_size = 0.2.
- t : contrôle la transparence des particules illustrées sur la figure. Par exemple, t = 0.2.

\* Modifie les unités de la position des particules afin qu'elle soit en kpc et non en 100 kpc.

\*\* Ce code n'a pas été écrit pour être exécutable sur Stargate puisqu'il est seulement utilisé pour visualiser rapidement une simulation à un temps précis (le code utilise 'matplotlib.pyplot.show()') et ne sauvegarde pas la figure).

## 1.20 gas\_density\_animation.py

*Auteur : Jérémie Lesage*

*Modifié par : Valérie Gendron*

Produit une animation de l'évolution de la densité de gaz de la simulation isolée 'I', 'I\_h' ou 'I\_hh' en utilisant 6 temps différents.

### Paramètres à modifier au besoin :

- name : nom de la simulation choisie (par exemple, name = 'I\_h')
- file1 à file6 : directories des fichiers de données 'g' correspondant aux temps auxquels on souhaite animer la densité de gaz.
- tsteps : liste contenant les temps en Ga auxquels on souhaite animer la densité de gaz.

\* Ce code n'a pas été écrit pour être exécutable sur Stargate.

## 1.21 global\_sfr\_profile.py

*Auteur : Valérie Gendron*

Calcule le taux de formation stellaire (SFR) global selon le temps et à différentes distances du centre de masse. Doit être utilisé avec un fichier 'sfr\_profile.out', lequel est obtenu en utilisant le code Fortran 'sfr\_profile.f'. Le code 'global\_sfr\_profile.py' utilise les SFR du fichier 'sfr\_profile.out' (taux de formation stellaire selon le temps dans différents anneaux concentriques) et les additionne pour obtenir le SFR total contenu à l'intérieur d'un certain rayon.

### Paramètres à modifier au besoin :

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'i\_h'). Doit aussi correspondre au nom du dossier contenant le fichier 'sfr\_profile.out'.
- galaxy\_type : soit 'impactor' ou 'target' (pour la galaxie cible). Permet de choisir des distances prédéterminées. Pour 'impactor', le profil global est obtenu aux distances 0.25, 1.25, 2.25, 3.25 et 4.25 kpc du centre de masse. Pour 'target', le profil global est obtenu aux distances 1.25, 2.25, 5.25, 10.25 et 19.75 kpc du centre de masse. Ces distances peuvent être changées manuellement si désiré.
- fileName : correspond au directory du fichier 'sfr\_profile.out' (par exemple, fileName = 'I/sfr\_profile.out').

## 1.22 g-s-masses.py

*Auteur : Jérémie Lesage*

*Modifié par : Valérie Gendron*

Trace la masse de gaz et de nouvelles étoiles selon le rayon pour différents temps. Le code utilise présentement les temps 0.02, 0.5, 1.0, 1.5, 2.0 et 3.0 Ga, mais ils peuvent être changés au besoin. Nécessite le fichier d'output 'sfr\_profile.out' obtenu avec le code 'sfr\_profile.f' pour exécuter.

### Paramètres à modifier au besoin :

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'i\_hh'). Doit correspondre au nom du dossier contenant le fichier d'output 'sfr\_profile.out'. Sinon, changer la variable 'fileName'.
- fileName : directory du fichier d'output 'sfr\_profile.out'.

## 1.23 gmap\_xy.py

*Auteur : Christian Carles*

*Modifié par : Valérie Gendron*

Permet d'obtenir des cartes de couleur de la densité de gaz dans le plan xy de la simulation choisie à 6 temps différents.

### Paramètres à modifier au besoin :

- time : liste contenant les 6 temps en Ga auxquels on veut obtenir des cartes de densité (par exemple, time = [0.0, 0.5, 1.0, 1.5, 2.0, 3.0]).
- fileName, fileName2, ..., fileName6 : noms des fichiers de données 'g' correspondant aux temps indiqués dans la liste 'time'.
- nbins : liste contenant le nombre de bins dans la direction x et dans la direction y des cartes (par exemple, nbins = [150, 150]).
- picName : nom de l'image obtenue.
- xcut et ycut : valeur maximale des axes x et y en unités de 100 kpc (par exemple, xcut = 0.05)
- lim : limite des axes x et y en kpc (par exemple, lim = 5 ; même chose que xcut et ycut).
- v\_min : valeur de densité minimale indiquée sur la barre de couleur.
- v\_max : valeur de densité maximale indiquée sur la barre de couleur.

\* Attention : il semble que les valeurs indiquées sur la barre de couleur ne sont pas exactes. Le code devrait être modifié. Cependant, il donne tout de même une idée générale de la répartition du gaz dans la galaxie.

## 1.24 gmap\_xz.py

*Auteur : Christian Carles*

*Modifié par : Valérie Gendron*

Même chose que 'gmap\_xy.py', mais pour le plan xz.

## 1.25 gmap\_yz.py

*Auteur : Christian Carles*

*Modifié par : Valérie Gendron*

Même chose que 'gmap\_yz.py', mais pour le plan yz.

## 1.26 H-O-Fe-masses.py

*Auteur : Jérémie Lesage*

*Modifié par : Valérie Gendron*

Permet d'obtenir deux figures : une contenant l'évolution temporelle des masses d'hydrogène, d'oxygène et de fer pour la simulation précisée et l'autre contenant la métallicité ( $[O/H]$  et  $[Fe/H]$ ) en fonction du temps. Le code utilise les fichiers d'output 'Z\_f.out' (obtenu avec le code fortran 'Z\_fibre'), 'Z\_g.out' ou 'Z\_h.out', et assume que 'Z\_f.out' prend en compte les particules à l'intérieur d'un rayon de 1 kpc du centre du système et que 'Z\_g.out' prend en compte les particules à l'intérieur d'un rayon de 10 kpc du centre du système.

### Paramètres à modifier au besoin :

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'I'). Doit correspondre au nom du dossier contenant le fichier d'output utilisé ('Z\_f', 'Z\_g' ou 'Z\_h'). Sinon, changer le premier argument de 'np.loadtxt()' pour qu'il corresponde au directory du fichier d'output utilisé.
- file\_choice : fichier d'output utilisé. Entrer 'f' pour 'Z\_f.out', 'g' pour 'Z\_g.out' ou 'h' pour 'Z\_h.out'.

\* Ce code n'a pas été écrit pour être exécutable sur Stargate.

## 1.27 metallicity\_profiles.py

*Auteur : Valérie Gendron*

Montre la métallicité ( $[O/H]$ ,  $[Fe/H]$  et  $[O/Fe]$ ) de la simulation choisie au temps 3.0 Gyr pour le gaz ou pour les étoiles. La métallicité est portée en graphique selon des anneaux concentriques à partir du centre de masse du système (anneaux de 0.5 kpc d'épaisseur, de 0 à 20 kpc). Permet aussi d'obtenir  $[O/Fe]$  selon  $[Fe/H]$ .

### Paramètres à modifier au besoin :

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'Af\_v1'). Doit correspondre au nom du dossier contenant les fichiers de données 'g' et 's'.
- dlim : limite en kpc du disque selon l'axe z (par exemple, dlim = 5). Le code ne tient compte que des particules ayant un z entre -dlim et +dlim.
- rlim : plus grand rayon [kpc] d'anneau considéré dans le plan xy du disque (par exemple, rlim = 20).
- content : choisir 'g' ou 's' pour obtenir le graphique des métallicités des particules de gaz ou d'étoiles, respectivement.
- fileName : (situé dans la première if clause et dans la première elif clause) directory du fichier 'g' et 's' correspondant au dernier temps de la simulation (par exemple, fileName = str(galaxy\_name) + "/g000564r" et fileName = str(galaxy\_name) + "/s000564r", ce qui pourrait par exemple donner Af\_v1/g000564r et Af\_v1/s000564r).

\* Le code a été écrit en assumant que les fichiers 'g' et 's' utilisés ont été préalablement modifiés avec 'detilt.f' afin que le moment cinétique soit en z seulement, que la position et la vitesse du centre de masse soient nulles, que les positions soient en unités de kpc et que les masses soient en unités de masses solaires.

## 1.28 mean\_O\_Fe-Fe\_H

*Auteur : Valérie Gendron*

Trace les profils de métallicité du gaz ou des étoiles de la simulation choisie au temps 3.0 Gyr. Très semblable au code 'metallicity\_profiles.py', mais permet aussi d'obtenir le [O/Fe] moyen dans chaque anneau selon le [Fe/H] ainsi que le [O/Fe] moyen dans chaque anneau selon le [Fe/H] moyen dans chaque anneau. Les graphiques obtenus sont des nuages de points avec un gradient de couleur (jaune = anneau près du centre, bleu = anneau vers l'extérieur).

### Paramètres à modifier au besoin :

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'Af\_v1'). Doit correspondre au nom du dossier contenant les fichiers de données 'g' et 's'.
- dlim : limite en kpc de l'épaisseur du disque selon l'axe z (par exemple, dlim = 5). Le code ne tient compte que des particules ayant un z entre -dlim et +dlim.
- rlim : plus grand rayon [kpc] d'anneau considéré dans le plan xy du disque (par exemple, rlim = 20).
- content : choisir 'g' ou 's' pour obtenir le graphique des métallicités des particules de gaz ou d'étoiles, respectivement.
- fileName : (situé dans la première if clause et dans la première elif clause) directory du fichier 'g' et 's' correspondant au dernier temps de la simulation (par exemple, fileName = str(galaxy\_name) + "/g000564r" et fileName = str(galaxy\_name) + "s000564r", ce qui pourrait par exemple donner Af\_v1/g000564r et Af\_v1/s000564r).

\* Le code a été écrit en assumant que les fichiers 'g' et 's' utilisés ont été préalablement modifiés avec 'detilt.f' afin que le moment cinétique soit en z seulement, que la position et la vitesse du centre de masse soient nulles, que les positions soient en unités de kpc et que les masses soient en unités de masses solaires.

## 1.29 metallicity\_bins\_Fe\_H

*Auteur : Valérie Gendron*

Trace le [O/Fe] moyen dans chaque intervalle de [Fe/H] pour les particules de gaz ou d'étoiles. Le code calcule le [O/Fe] et le [Fe/H] de chaque particule (de gaz ou d'étoiles). Il sépare ensuite les valeurs de [Fe/H] en intervalles de 0.02 de largeur. Pour toutes les particules dont le [Fe/H] fait partie d'un intervalle particulier, le [O/Fe] est considéré. On fait ensuite la moyenne du [O/Fe] des particules appartenant à chaque intervalle de [Fe/H].

### Paramètres à modifier :

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'Af\_v1'). Doit correspondre au nom du dossier contenant les fichiers de données 'g' et 's'.
- dlim : limite en kpc du disque selon l'axe z (par exemple, dlim = 5). Le code ne tient compte que des particules ayant un z entre -dlim et +dlim.
- rlim : plus grand rayon [kpc] d'anneau considéré dans le plan xy du disque (par exemple, rlim = 20).
- width : largeur des intervalles de [Fe/H] (par exemple, width = 0.02).
- content : choisir 'g' ou 's' pour obtenir le graphique des métallicités des particules de gaz ou d'étoiles, respectivement.
- fileName : (situé dans la première if clause et dans la première elif clause) directory du fichier 'g' et 's' correspondant au dernier temps de la simulation (par exemple, fileName = str(galaxy\_name) + "/g000564r" et fileName = str(galaxy\_name) + "s000564r", ce qui pourrait par exemple donner Af\_v1/g000564r et Af\_v1/s000564r).

\* Le code a été écrit en assumant que les fichiers 'g' et 's' utilisés ont été préalablement modifiés avec 'detilt.f' afin que le moment cinétique soit en z seulement, que la position et la vitesse du centre de masse soient nulles, que les positions soient en unités de kpc et que les masses soient en unités de masses solaires.

### 1.30 rotation\_curves\_comparison.py

*Auteur : Valérie Gendron*

Permet de comparer les courbes de rotation de différentes simulations pour les particules de gaz ou d'étoiles à un temps précisé (code initialement écrit pour  $t=3.0$  Ga).

**Paramètres à modifier au besoin :**

- `content` : choisir 'g' ou 's' pour obtenir le graphique des courbes de rotation des particules de gaz ou d'étoiles, respectivement.
- `time` : temps en Ga auquel on veut obtenir les courbes de rotation.
- `zlim` : limite en  $z$  des particules incluses dans les courbes de rotation. Seules les particules dont  $-zlim \leq z \leq +zlim$  sont considérées.
- `rlim` : seules les particules dont  $0 \leq \sqrt{x^2 + y^2} < rlim$  sont incluses dans les courbes de rotation.
- `galaxy_names` : liste contenant le nom de toutes les simulations qu'on souhaite comparer (par exemple, `galaxy_name = ['I', 'Af_v1', 'Ar_v1', 'As_v1']`).

\* Le code a été écrit en assumant que les fichiers 'g' et 's' utilisés ont été préalablement modifiés avec 'detilt.f' afin que le moment cinétique soit en  $z$  seulement, que la position et la vitesse du centre de masse soient nulles, que les positions soient en unités de kpc et que les masses soient en unités de masses solaires.

\*\* Utilise le fichier 'tzstep.dat'.

### 1.31 separation\_for\_comparison.py

*Auteur : Valérie Gendron*

Sépare manuellement la totalité des particules d'étoiles et des particules de gaz à des temps précis et réécrit des fichiers 's' et 'g' ne contenant que les particules de la galaxie cible dans un dossier nommé 'data\_target' (doit être créé manuellement avant d'exécuter le code).

**Paramètres à modifier au besoin :**

- `galaxy_name` : assigner le nom de la simulation désirée (par exemple, `galaxy_name = 'Af_v1'`). Doit aussi correspondre au nom du dossier contenant les fichiers de données brutes 's' de la simulation.
- `sfileNames1, sfileNames2, ...` : listes contenant les noms des fichiers 's' à séparer (par exemple, `sfileNames1 = ['s000005', 's000007', 's000010']`). Chaque liste contient les noms des fichiers qui seront séparés avec la même limite.
- `gfileNames1, gfileNames2, ...` : listes contenant les noms des fichiers 'g' à séparer (par exemple, `gfileNames1 = ['g000005', 'g000007', 'g000010']`). Chaque liste contient les noms des fichiers qui seront séparés avec la même limite.
- `conditions` des boucles 'for  $i$  in range(0, len(c0))' : assigner une limite de position permettant de séparer les particules d'étoiles ou de gaz entre la galaxie cible et l'impacteur. Par exemple, 'if  $c0[i] > -0.10$ ' signifie que toutes les particules dont  $x > -10$  kpc font partie de la galaxie cible ( $c0$  contient les positions en  $x$  en unités de 100 kpc,  $c1$  contient les positions en  $y$  en unités de 100 kpc et  $c2$  contient les positions en  $z$  en unités de 100 kpc).

### 1.32 sfr\_collisions.py

*Auteur : Valérie Gendron*

Montre le taux de formation stellaire selon le temps pour l'ensemble du système de la simulation précisée.

**Paramètres à modifier au besoin :**

- `galaxy_name` : assigner le nom de la simulation désirée (par exemple, `galaxy_name = 'Af_v1'`). Doit aussi correspondre au nom du dossier contenant les fichiers de données brutes 's' de la simulation, ainsi que le fichier 'tzstep.dat'.

\* Modifie les unités de masse et de temps afin d'afficher le taux de formation stellaire en unités de masses solaires par an.

### 1.33 sfr\_profile.py

*Auteur : Valérie Gendron*

Trace le taux de formation stellaire de la simulation précisée selon le temps à différentes distances du centre. Le code utilise les taux de formation stellaire contenus dans le fichier d'output 'sfr\_profile.out', obtenu avec le code Fortran 'sfr\_profile.f' (ces taux de formation stellaire sont calculés dans des anneaux concentriques à partir du centre).

#### Paramètres à modifier :

- galaxy\_name : nom de la simulation désirée.
- galaxy\_type : choisir 'target' ou 'impactor'. Lorsqu'on choisit 'target', le SFR selon le temps est tracé à des distances de 1.25, 2.25, 5.25, 10.25 et 19.75 kpc du centre de masse, alors que lorsqu'on choisit 'impactor', le SFR selon le temps est tracé à des distances de 0.25, 1.25, 2.25, 3.25 et 4.25 kpc du centre de masse.
- fileName : directory du fichier d'output 'sfr\_profile.out' (par exemple, fileName = str(galaxy\_name) + "/sfr\_profile.out").

### 1.34 smap.py

*Auteur : Christian Carles*

*Modifié par : Valérie Gendron*

Permet d'obtenir des cartes de couleur de la densité d'étoiles de la simulation choisie à 6 temps différents.

#### Paramètres à modifier au besoin :

- plan : choisir 'xy', 'xz' ou 'yz' pour obtenir les densités d'étoiles dans le plan 'xy', 'xz' ou 'yz' respectivement.
- dim : dimension de la carte en kpc. Par exemple, si dim = 5, la carte a une dimension de 5 kpc x 5 kpc.
- time : liste contenant les 6 temps en Ga auxquels on veut obtenir des cartes de densité (par exemple, time = [0.0, 0.5, 1.0, 1.5, 2.0, 3.0]).
- fileName, fileName2, ..., fileName6 : noms des fichiers de données 's' correspondant aux temps indiqués dans la liste 'time'.
- v\_min : valeur de densité minimale indiquée sur la barre de couleur.
- v\_max : valeur de densité maximale indiquée sur la barre de couleur.
- nbins : liste contenant le nombre de bins dans les deux directions du plan choisi (par exemple, nbins = [150, 150]).

\* Attention : il semble que les valeurs indiquées sur la barre de couleur ne sont pas exactes. Le code devrait être modifié. Cependant, il donne tout de même une idée générale de la répartition des étoiles dans la galaxie.

### 1.35 time\_metallicity\_profiles.py

*Auteur : Valérie Gendron*

Trace les profils de métallicité selon le temps dans des anneaux à 1 et à 10 kpc du centre de masse du système. Produit une première figure contenant [O/H], [Fe/H] et [O/Fe] selon le temps à 1 kpc, puis une deuxième figure contenant [O/H], [Fe/H] et [O/Fe] selon le temps à 10 kpc, et enfin une troisième figure contenant [O/Fe] selon [Fe/H] à 1 et à 10 kpc.

#### Paramètres à modifier au besoin :

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'Af\_v1'). Doit correspondre au nom du dossier contenant les fichiers de données 'g' et 's'.
- content : choisir 'g' ou 's' pour obtenir le graphique des métallicités des particules de gaz ou d'étoiles, respectivement.
- dlim : limite en kpc de l'épaisseur du disque selon l'axe z (par exemple, dlim = 5). Le code ne tient compte que des particules ayant un z entre -dlim et +dlim.
- rlim : plus grand rayon [kpc] d'anneau considéré dans le plan xy du disque (par exemple, rlim = 20).

\* Le code a été écrit en assumant que les fichiers 'g' et 's' utilisés ont été préalablement modifiés avec 'detilt.f' afin que le moment cinétique soit en z seulement, que la position et la vitesse du centre de masse soient nulles, que les positions soient en unités de kpc et que les masses soient en unités de masses solaires.

\*\* Utilise le fichier 'tzstep.dat'.

### 1.36 velocity\_dispersion.py

*Auteur : Valérie Gendron*

Trace les dispersions des vitesses en coordonnées cylindriques selon la distance au centre de masse du système pour les particules de gaz ou d'étoiles à différents temps. Le fichier contient deux fonctions : 'velocity\_dispersion', qui produit autant de figures qu'il y a de temps différents dans les fichiers d'output 'sigma\_vgc.out' ou 'sigma\_vsc.out' utilisés et trace les trois dispersions de vitesses ( $\sigma_r$ ,  $\sigma_t$ ,  $\sigma_z$ ) sur le même graphique, et 'velocity\_dispersion\_time', qui produit trois figures, une pour chaque direction de vitesse (radiale, tangentielle et en z); tous les temps précisés sont représentés sur chacune des figures.

#### Paramètres à modifier au besoin :

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'Af\_v1').
- time\_list : liste contenant les temps en Ga auxquels les graphiques de dispersions de vitesses seront obtenus (par exemple, time\_list = [0.3, 0.48, 1.0, 2.0, 3.0]). Les temps doivent se retrouver dans le fichier d'output 'sigma\_vgc.out' ou 'sigma\_vsc.out'.
- content : choisir 'g' ou 's' pour obtenir les dispersions de vitesses des particules de gaz ou d'étoiles, respectivement.
- fileName : nom du directory du fichier d'output 'sigma\_vgc.out' ou 'sigma\_vsc.out' à utiliser.

\* Les fichiers d'output 'sigma\_vgc.out' et 'sigma\_vsc.out' sont obtenus en utilisant les codes Fortran 'sigma\_Vg.f' et 'sigma\_Vs.f'.

### 1.37 velocity\_dispersion\_animation.py

*Auteur : Valérie Gendron*

Fait une animation des dispersions de vitesses d'une simulation pour tous les temps excepté 0. Les points rouges, verts et bleus représentent respectivement les dispersions de vitesses radiale ( $\sigma_r$ ), tangentielle ( $\sigma_t$ ) et selon z ( $\sigma_z$ ). Utilise les fichiers d'output 'sigma\_vgc.out' ou 'sigma\_vsc.out' (obtenus avec les codes Fortran 'sigma\_Vg.f' et 'sigma\_Vs.f') et le fichier 'tzstep.dat' correspondant.

#### À faire avant de pouvoir exécuter le code (sur Stargate) :

Voir la section 'collision\_animation\_g-s.py' de ce document.

#### Paramètres à modifier au besoin :

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'Af\_v1').
- content : choisir 'g' ou 's' pour obtenir les dispersions de vitesses des particules de gaz ou d'étoiles, respectivement.
- fileName : nom du directory du fichier d'output sigma à utiliser.

### 1.38 velocity\_disp\_and\_evol\_animation.py

*Auteur : Valérie Gendron*

Produit une animation séparée en quatre sous-figures : dispersions de vitesses radiales, tangentielles et verticales des particules de gaz (en haut à gauche) et des particules d'étoiles (en bas à gauche), et évolution des particules de gaz et d'étoiles dans le plan xz (en haut à droite) et dans le plan xy (en bas à droite), selon le temps. Les points rouges, verts et bleus des sous-figures de gauche représentent respectivement les dispersions de vitesses radiales ( $\sigma_r$ ), tangentielles ( $\sigma_t$ ) et verticales ( $\sigma_z$ ). Les points bleu foncé, bleu pâle, jaunes et rouges représentent respectivement les particules d'étoiles de la galaxie cible, les particules de gaz de la galaxie cible, les particules d'étoiles de l'impacteur et les particules de gaz de l'impacteur. Ce code est une combinaison des codes 'collision\_animation\_g-s.py' et 'velocity\_dispersion\_animation.py'. Il utilise les fichiers d'output 'sigma\_vgc.out', 'sigma\_vsc.out' (obtenus avec les codes Fortran 'sigma\_Vg.f' et 'sigma\_Vs.f') ainsi que les fichiers 'gxxxxx\_1', 'gxxxxx\_2', 'sxxxxx\_1', 'sxxxxx\_2' (obtenus avec le code Fortran 'split.f') et 'tzstep.dat' correspondants.

#### À faire avant de pouvoir exécuter le code (sur Stargate) :

Voir la section 'collision\_animation\_g-s.py' de ce document.



### Paramètres à modifier au besoin :

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'Af\_v1').
- fileName (dans la fonction 'velocity\_dispersion') : nom du directory des fichiers d'output sigma à utiliser.
- time\_list : liste contenant tous les temps qui sont inclus dans l'animation. S'assurer que la directory du fichier 'tzstep.dat' est la bonne.

\*Le code ne fonctionne pas parfaitement et nécessite des modifications.

## 1.39 velocity\_distribution\_comparison.py

*Auteur : Valérie Gendron*

Permet de comparer les distributions de vitesses tangentielles ou radiales pour les particules de gaz et d'étoiles de plusieurs simulations à un temps précisé (code initialement écrit pour  $t=3.0$  Ga).

### Paramètres à modifier au besoin :

- content : choisir 'g' ou 's' pour obtenir les distributions de vitesses des particules de gaz ou d'étoiles, respectivement.
- velocity : choisir 't' ou 'r' pour obtenir les distributions de vitesses tangentielles ou radiales, respectivement.
- time : temps en Ga auquel on veut obtenir les dispersions de vitesses.
- width : largeur en km/s des bins de vitesse qui seront utilisés pour les distributions de vitesses.
- rlim : seules les particules dont  $0 \leq \sqrt{x^2 + y^2} < \text{rlim}$  sont incluses dans les distributions de vitesses.
- zlim : seules les particules dont  $-\text{zlim} \leq z \leq +\text{zlim}$  sont incluses dans les distributions de vitesses.
- galaxy\_names : liste contenant le nom de toutes les simulations qu'on souhaite comparer (par exemple, galaxy\_name = ['I', 'Af\_v1', 'Ar\_v1', 'As\_v1']).

\* Le code a été écrit en assumant que les fichiers 'g' et 's' utilisés ont été préalablement modifiés avec 'detilt.f' afin que le moment cinétique soit en z seulement, que la position et la vitesse du centre de masse soient nulles, que les positions soient en unités de kpc et que les masses soient en unités de masses solaires.

\*\* Utilise le fichier 'tzstep.dat'.

## 1.40 warp.py

*Auteur : Valérie Gendron*

Trace la déformation du disque de la simulation précisée en fonction de la phase, au temps 3.0 Gyr et à deux rayons différents.

### Paramètres à modifier au besoin :

- galaxy\_name : assigner le nom de la simulation désirée (par exemple, galaxy\_name = 'Af\_v1'). Doit correspondre au nom du dossier contenant les fichiers de données 'g' et 's'.
- content : choisir 'g' ou 's' pour obtenir la déformation du gaz ou des étoiles, respectivement.
- fileName : (situé dans la première if clause et dans la première elif clause) nom du fichier d'output (obtenu avec le code Fortran 'warp.f') contenant les phases, les moyennes de position z, les dispersions, et le nombre de particules pour un certain intervalle de rayons. Par exemple, fileName = "g000564r\_5-6\_warp.out" et fileName = "s000564r\_5-6\_warp.out".
- fileName2 : même chose que pour fileName, mais indiquer le nom du fichier d'output pour le deuxième intervalle de rayon considéré. Par exemple, fileName2 = "g000564r\_10-11\_warp.out" et fileName2 = "s000564r\_10-11\_warp.out".

## 2 Codes Fortran

### 2.1 detilt.f

*Auteur : Simon Richard*

*Modifié par : Hugo Martel*

Prend en input les fichiers de données brutes 'dxxxxxx', 'gxxxxxx' et 'sxxxxxx' (pour une dump particulière, par exemple 'd000105', 'g000105' et 's000105') et les réécrit dans des fichiers 'dxxxxxxr', 'gxxxxxxr' et 'sxxxxxxr' en corrigeant les positions (x, y, z) et les vitesses (vx, vy, vz) afin que la position et la vitesse du centre de masse du système soient nulles et pour que le moment cinétique du système soit normalisé en z (0,0,1). Le code modifie aussi les unités pour que les positions soient en kpc, les vitesses en km/s et les masses en masses solaires.

### 2.2 sfr\_profile.f

*Auteur : Christian Carles*

Permet d'obtenir le temps et la position de chaque événement de formation stellaire en identifiant les particules de gaz qui se transforment en particules d'étoiles. Le code produit un fichier d'output nommé 'sfr\_profile.out' contenant :

- Colonne 1 : temps;
- Colonne 2 : distance du bin;
- Colonne 3 : épaisseur du bin;
- Colonne 4 : masse de gaz dans le bin;
- Colonne 5 : masse d'étoiles qui étaient du gaz au temps précédent dans le bin;
- Colonne 6 : SFR (colonne 5 divisée par dt, qui est fixe);
- Colonne 7 : SSFR (SFR divisé par la masse de gaz dans le bin; ne fonctionne pas très bien);
- Colonne 8 : SFR accumulé dans les bins jusqu'à ce bin;
- Colonne 9 : numéro du bin.

\* Utilise le fichier tzstep.dat.

### 2.3 sigma\_Vg.f

*Auteur : Christian Carles*

*Modifié par : Valérie Gendron*

Permet de calculer, pour chaque intervalle de temps, dans des bins radiaux, la dispersion de vitesse des particules de gaz. Le calcul est fait en utilisant les coordonnées cylindriques (r,  $\theta$ , z), mais peut être modifié facilement pour utiliser les coordonnées cartésiennes (x, y, z). Le code produit un fichier d'output nommé 'sigma\_vgc.out' contenant :

- Colonne 1 : temps;
- Colonne 2 : rayon central du bin;
- Colonne 3 : largeur du bin;
- Colonne 4 : numéro du bin;
- Colonne 5 : nombre de particules de gaz dans le bin;
- Colonne 6 :  $\sigma_r$  (dispersion de vitesse radiale) des particules dans le bin;
- Colonne 7 :  $\sigma_t$  (dispersion de vitesse tangentielle) des particules dans le bin;
- Colonne 8 :  $\sigma_z$  (dispersion de vitesse verticale) des particules dans le bin.

**Paramètres à modifier au besoin :**

- rmax : rayon maximal atteint.
- nbins : nombre total de bins à utiliser.

\* Prend en compte les particules de feedback.

\*\* Utilise le fichier tzstep.dat.

## 2.4 sigma\_Vs.f

*Auteur : Christian Carles*

*Modifié par : Valérie Gendron*

Même chose que 'sigma\_Vgc.f', mais pour les particules d'étoiles. Le fichier d'output est nommé 'sigma\_vsc.f'.

\* Ne prend pas en compte les particules de feedback.

\*\* Utilise le fichier tzstep.dat.

## 2.5 split.f

*Auteur : Hugo Martel*

Prend en input les fichiers de données brutes 'dxxxxxx', 'gxxxxxx' et 'sxxxxxx' (pour une dump particulière, par exemple 'd000105', 'g000105' et 's000105') et sépare les données contenues dans chaque fichier en les réécrivant dans deux nouveaux fichiers (les fichiers originaux ne sont pas modifiés). Les noms des nouveaux fichiers sont les noms des fichiers originaux avec un '\_1' ou un '\_2' ajouté (par exemple, 's000105\_1' et 's000105\_2'). Les fichiers finissant par '\_1' contiennent les particules de la galaxie cible et les fichiers finissant par '\_2' contiennent les particules de l'impacteur.

### Paramètres à modifier au besoin :

- ligne "parameter (nm=1000000)" : taille du tableau où les données sont inscrites. Pour les simulations à très haute résolution, la taille (nm= ) devrait être augmentée.

### Simulations à basse résolution :

- if(idg(i).lt.128000) then
- if(pn(i).lt.320000) then

### Simulations à haute résolution

- if(idg(i).lt.256000) then
- if(pn(i).lt.640000) then

### Simulations à très haute résolution

- if(idg(i).lt.512000) then
- if(pn(i).lt.1280000) then

\* Ne modifie pas les unités. Les positions sont encore en unités de 100 kpc, les vitesses en unités de 207.4 km/s, les masses en unités de  $10^{12} M_{\text{sun}}$ , etc.

\*\* Utilise les numéros initialement attribués aux particules pour les séparer. Cependant, pendant la majorité des simulations de collisions, les particules changent de galaxie et le split n'est donc plus exact.

## 2.6 warp.f

*Auteur : Hugo Martel*

Permet de calculer la déformation du disque (gaz ou étoiles) d'une simulation à un temps précis. Le programme demande le nom d'un fichier d'entrée (par exemple, s000564r), puis un rayon minimum et un rayon maximum. Il calcule alors la moyenne et la dispersion des valeurs de  $z$  dans des secteurs le long de l'anneau. Le fichier d'output est nommé automatiquement (par exemple, s000564r\_warp.out). Les colonnes contiennent :

- Colonne 1 : phase (entre 0 et  $2\pi$ );
- Colonne 2 : moyenne;
- Colonne 3 : dispersion;
- Colonne 4 : nombre de particules.

### Paramètres à modifier au besoin :

- `nwedge` : nombre de secteurs (phase). S'assurer qu'il y ait assez de particules par secteur pour ne pas que les résultats soient trop bruités. Sinon, réduire le nombre de secteurs utilisés pour qu'il y ait plus de particules par secteur.
- `zmax` : limite en  $z$  de la position des particules considérées dans le calcul de la déformation du disque.

\* Le code suppose que le programme 'detilt' a déjà été exécuté.

\*\* Phase = 0 =  $2\pi$  correspond à  $x = 0, y > 0$  et phase =  $\pi$  correspond à  $x < 0, y = 0$ .

## 2.7 Z\_fibre.f

*Auteur : Christian Carles*

Calcule l'évolution de l'abondance des éléments à l'intérieur d'un rayon donné. Le code produit un fichier d'output nommé 'Z\_f.out' contenant :

- Colonne 1 : temps;
- Colonne 2 : nombre de particules d'étoiles dans la région;
- Colonne 3 : nombre de particules de gaz dans la région;
- Colonnes 4-9 : pour H [total mass, input mass, output mass, mass consumed in SF, mass from SN feedback, total mass variation];
- Colonnes 10-15 : pour O [total mass, input mass, output mass, mass consumed in SF, mass from SN feedback, total mass variation];
- Colonne 16 :  $[O/H]$ ;
- Colonnes 17-22 : pour Fe [total mass, input mass, output mass, mass consumed in SF, mass from SN feedback, total mass variation];
- Colonne 23 :  $[Fe/H]$ ;
- Colonne 24-29 : pour le gaz total [total mass, input mass, output mass, mass consumed in SF, mass from SN feedback, total mass variation];
- Colonne 30 : masse totale d'étoiles;
- Colonne 31 : densité moyenne  $\rho$ .

Le paramètre 'rad' détermine à l'intérieur de quel rayons la métallicité est calculée et il peut être modifié au besoin.

\* Utilise le fichier tzstep.dat.