



HAL
open science

Identification et analyse des structures stellaires émergeant des régions de formation d'étoiles

Timothé Roland

► **To cite this version:**

Timothé Roland. Identification et analyse des structures stellaires émergeant des régions de formation d'étoiles. Astrophysique [astro-ph]. Université de Strasbourg, 2021. Français. NNT : 2021STRAE005 . tel-03653005

HAL Id: tel-03653005

<https://theses.hal.science/tel-03653005>

Submitted on 27 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ÉCOLE DOCTORALE PHYSIQUE, CHIMIE-PHYSIQUE
UMR 7550 - Observatoire Astronomique de Strasbourg

THÈSE

présentée par :

Timothé ROLAND

soutenue le : 22 janvier 2021

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/ Spécialité : Astrophysique

Identification et analyse des structures stellaires émergeant des régions de formation d'étoiles

THÈSE dirigée par :

M. BOILY Christian

M. CAMBRÉSY Laurent

Professeur des universités, Université de Strasbourg - ObAS

Astronome, Université de Strasbourg - ObAS

RAPPORTEUSES :

Mme JONCOUR Isabelle

Mme MORAUX Estelle

Maîtresse de conférences, Université Grenoble-Alpes - IPAG

Maîtresse de conférences, Université Grenoble-Alpes - IPAG

AUTRES MEMBRES DU JURY :

M. FAMAHEY Benoît

M. BOILY Christian

M. BIENAYMÉ Olivier

M. MONTILLAUD Julien

Directeur de recherche, CNRS - ObAS

Professeur, Université de Strasbourg - ObAS

Astronome, Université de Strasbourg - ObAS

Maître de conférences, Université de Franche-Comté - UTINAM

REMERCIEMENTS

Quel enfer de devoir lister toutes les personnes qui ont concouru à cette thèse ! Car oui la thèse n'est pas juste un épais manuscrit incompréhensible pour le commun des mortels (d'ailleurs pas si épais que ça dans mon cas, et j'espère relativement compréhensible...). La thèse a été pour moi une vraie expérience de vie. Pendant ces trois ans, mon travail de recherche et ma vie personnelle ne faisait plus trop de différences, à ma plus grande joie au début, peut-être un peu trop à mes dépens vers la fin... J'ai beaucoup appris sur moi mais j'ai surtout rencontré des gens merveilleux ici à Strasbourg.

Je remercie donc en premier lieu ceux qui m'ont lancé dans cette environnement de recherche, à savoir mes deux encadrants : Christian et Laurent. J'ai l'impression de me reconnaître en chacun de vous, Christian avec ton esprit en escalier et toujours très, des fois trop, passionné par tes travaux, et Laurent plus discret mais qui ne cessait d'apporter de la mesure à nos discussions. Merci pour votre soutien, votre disponibilité et l'opportunité que vous m'avez donné.

Je dois ensuite mentionner la bande de copains des Bablets que j'ai longuement appris à connaître depuis l'observatoire jusqu'aux tables du *Télégraphe* où nous exaltions tous ensemble. J'ai nommé Joe mon cobureau dont la compagnie était toujours appréciée ; Paolo arrivé plus tard mais juste à temps pour m'épauler aussi bien dans mes recherches que dans la vie nocturne strasbourgeoise ; Jéjé, Mathieu, François pour les chants plurivocaux ; Jon et Julie pour les afters ; Nicolas et Lucie pour la passion de la bière ; encore Nicolas pour les idées Monte Carlo ; Julien pour ta thèse indispensable à la mienne ; JB, pour ton éloge de la lenteur ; petit Nico pour les futsals... Je pourrais continuer longuement je pense, avec toutes les personnes de l'observatoire avec qui j'ai partagé des moments forts appréciables, Ariane, Benoît, Christophe, Ada, Lorenzo, Anaïs, Thomas, Katarina, Oliver, Yelena les deux Pierre, Dominique, Rodrigo, Évelyne, Magalie... Il faut en effet savoir qu'il règne une ambiance formidable à l'observatoire que ce soit avec les secrétaires, Post-Doc, documentalistes, informaticiens, chercheurs, ingénieurs. Je ne pourrais tous vous citer mais merci pour les discussions, les services rendus, les pauses café ou au pub, les parties de pétanques, le solstice d'été bien sûr et tout ce qui fait de l'observatoire un endroit magnifiquement vivant.

J'ai aussi une pensée pour les personnes du jardin des sciences qui m'ont appris l'art et l'envie de vulgariser ma science. J'espère que ce manuscrit est la preuve que j'ai bien appris qu'elle est abordable à tous. Milène, Natacha, Jean-Yves, Dorothée, Benoît, Marion, bonne chance pour le nouveau planétarium qui arrive !

Je tiens aussi à remercier toutes les personnes qui m'ont aidé à concrétiser ce parcours que ce soit les membres du jury avec lesquelles il est toujours agréable d'échanger, les autres rencontres faites en conférence, mon maître de stage Frédéric Galliano qui m'a propulsé dans le monde de la recherche mais aussi plus largement les profs de physique de TPS.

En parlant de TPS, comment oublier tous les amis qui au fil de mes études à Strasbourg, à Grenoble et à Lyon ont constitué la personne que je suis aujourd'hui. Aux militants du Génépi aussi à qui je dois ma conscience politique. Je pense tous bien à vous, car je pouvais compter sur chacun d'entre vous pour m'évader tout en

restant les pieds sur terre.

Enfin je remercie tout particulièrement mes parents ainsi que ma sœur et mon frère qui n'ont jamais remis en cause mes perspectives, m'ont accompagné dans mes choix mais aussi dans des épreuves bien moins joyeuses. À ce titre, j'ai une pensée émue pour tous les soignant-es qui m'ont permis tout simplement de continuer d'être là. Merci aussi à la grande famille, les oncles, tantes, cousins, j'aurais beaucoup aimé fêter cette thèse avec vous tous!

Pour finir, comment oublier celle qui m'a accompagné tout au long de cette thèse. Katell, tu as peut-être subie un peu plus mes peines que mes joies durant ces années et tu m'as supporté pendant un long confinement. Sans toi, j'aurais été bien démuni pour arriver au bouts de ces travaux. J'espère qu'on pourra partager maintenant nos meilleurs moments.

« Ah! Saloperie!!! »

Timothé Roland,

22-01-2021 à 14h 36min 45sec

TABLE DES MATIÈRES

Introduction	1
1.1 Conditions initiales de la formation d'étoiles	2
1.2 Processus physiques de la formation d'étoiles	3
1.2.1 Effondrement et fragmentation	3
1.2.2 Formation d'une proto-étoile	5
1.2.3 Fin de la formation d'étoiles	7
1.3 Amas d'étoiles	8
1.4 Approche numérique de la formation d'étoiles	9
1.5 La plateforme AMUSE	11
1.6 Plan de thèse	12
2 Modéliser un amas stellaire avec AMUSE	19
2.1 Introduction à AMUSE	19
2.2 Le problème à N -corps	21
2.2.1 Schémas d'intégration	21
2.2.2 Unités Hénon et équilibre du viriel	23
2.3 Tests des codes gravitationnels	24
2.3.1 Mise en place de conditions initiales	24
2.3.2 Intégration d'un modèle de Plummer à l'équilibre	25
2.3.3 Intégration d'une binaire excentrique	27
2.3.4 Interactions fortes au sein d'un amas	28
2.4 Conclusion	29
3 Identification de structures au sein d'amas fragmentés	31
3.1 Modèle de fragmentation aux N -corps	31
3.2 Algorithme de détection de surdensités	34
3.2.1 L'algorithme HOP	34
3.2.2 L'algorithme MST	38
4 Effets de projection dans les régions de formation stellaire [Article]	45
5 Propriétés dynamiques des groupes produits par les modèles GDF	57
5.1 Rapport masse/lumière η	57
5.1.1 Comparaison avec des groupes identifiés dans <i>Gaia</i> DR2	63

5.2	Cinématique des groupes	64
6	Évolution des populations de binaires au sein d’environnements denses	67
6.1	Observations	67
6.2	Simulations	69
6.2.1	Approche AMUSienne : le module <code>Multiple</code> s	70
6.2.2	Traitement des perturbations par le module <code>Multiple</code> s	72
6.3	Applications	74
7	Conclusion et perspectives	79
7.1	Incomplétude	79
7.2	Intégration d’un environnement gazeux au sein d’une binaire	81
7.3	Contexte galactique	82
A	Examples of AMUSE scripts	i
A.1	Simple setup of an N -body simulation	i
A.1.1	Set the initial conditions	i
A.1.2	Set the integrator	iv
A.1.3	Make the computation and store the results	v
A.1.4	End the script	v
A.1.5	Summary	vi
A.2	Further manipulation of an AMUSE particle set	viii
A.3	Example script to build a GDF model	ix
A.3.1	Using only a direct N -body model	ix
A.3.2	Using the <code>Multiple</code> s module	xvi

INTRODUCTION

La lumière des étoiles a longtemps constitué le seul moyen nous permettant de sonder les phénomènes physiques à l'œuvre au delà de la simple portée humaine. Ce rôle que jouent les étoiles dans notre univers est fondamental pour les astrophysiciens. Pourtant elles n'existent pas de toute éternité. Comme tout processus physique, elles ne constituent qu'une étape dans l'enchaînement de transformations que la matière, dite baryonique¹, subit. Suite au Big Bang, cette matière primordiale composée majoritairement d'hydrogène ne représentaient qu'un océan de gaz obscur. La transformation de ce gaz en étoiles a radicalement modifié la structure de l'univers.

On pourrait définir une étoile comme un réservoir de gaz d'hydrogène suffisamment dense d'environ 10^{27} atomes/cm³ (soit 1 kg/cm³), pour permettre la réaction de fusion thermonucléaire de ce dernier. C'est bien cette dernière caractéristique qui rend les étoiles si particulières. En effet, cette réaction est à l'origine de la diversité des éléments constitutifs de la matière. De plus, cette réaction libère une quantité d'énergie faramineuse à l'origine de leur lumière éclatante qui a conduit entre autre à ré-ioniser l'univers. Face au vide (relatif) de l'espace interstellaire, ayant une densité moyenne dans notre galaxie de seulement 1 atome/cm³, l'existence d'un tel objet semble hautement improbable. Pourtant notre galaxie, la Voie Lactée, en contient quelques centaines de milliards. La formation d'étoiles représente ainsi la suite de processus consistant à confiner le gaz diffus du milieu interstellaire (MIS) en l'objet compact qu'est une étoile. De la taille d'une galaxie à celle d'une étoile, l'ensemble de ce phénomène traverse plus de 10 ordres de grandeur et est au croisement de nombreux domaines de l'astrophysique.

Mon travail de thèse s'est concentré à la fin de ce processus, lorsque les jeunes étoiles émergent en petits groupes du nuage de gaz dont elles sont issues. Ces derniers comportent quelques centaines à milliers d'étoiles distribuées selon des configurations variées et évoluant de manière rapide et chaotique. Je me suis en particulier intéressé aux processus dynamiques à N -corps régissant l'évolution de ces groupes par le biais de modélisations numériques. Grâce à ces modèles, j'ai aussi pu tester des outils d'identification de ces groupes dans le but d'extraire leurs propriétés dynamiques dans des conditions similaires aux observations. Mieux contraindre ces propriétés permet de démêler les phénomènes qui en sont responsable ; soit issus directement du processus de formation d'étoiles ou, de l'évolution chaotique de ces groupes. Cela ouvre aussi la voie à l'étude à moyen et long terme des étoiles en amas liés ou se dissolvant dans le disque galactique.

Pour commencer, j'exposerai brièvement les étapes marquant le début du processus de formation d'étoiles, en se basant principalement sur la compréhension de ces phénomènes au sein de notre galaxie.

1. la matière baryonique représente la matière qui nous est familière et qui se présente sous la forme d'atomes composés de protons, de neutrons et d'électrons.

1.1 Conditions initiales de la formation d'étoiles

Toutes les galaxies contiennent des étoiles. En revanche seulement certaines sont encore capable d'en former. En effet le processus de formation stellaire commence par le réservoir de gaz qu'elles contiennent. Il est la condition nécessaire pour former des étoiles. Pourtant dans notre galaxie, la Voie Lactée, il représente seulement 10% de la masse des étoiles. La reconnaissance du rôle clé que joue le MIS au sein des galaxies est d'un intérêt relativement récent en lien avec l'amélioration des techniques d'observations. En cette année 2020, on pourra notamment se référer aux importants travaux de Françoise Combes, médaille d'or du CNRS de cette année, pour sa contribution dans ce domaine; avec p. ex. la détection de la molécule de monoxyde de carbone (CO) comme traceur des régions de formation d'étoiles au sein des disques de galaxies proches (Combes et al., 1977), ou encore des premières simulations de galaxies comportant un disque gazeux (Bournaud & Combes, 2002; Combes, 2004).

Le gaz du MIS est naturellement composé de l'élément le plus abondant de l'univers, l'hydrogène. On distingue généralement trois phases correspondant à ses états moléculaire, atomique et ionisé. Ces différentes phases ne représentent néanmoins que l'état du gaz au sein d'un continuum de conditions environnementales allant de régions denses et froides à des régions plus diffuses et ionisées. En effet, la proximité plus ou moins marquée du rayonnement des étoiles a pour effet de modifier les équilibres thermo-dynamiques du gaz. On remarque en revanche que la majeure partie du gaz capable en principe de remplir l'entièreté du volume de la galaxie ne se concentre en fait qu'en un disque fin. Le gaz du MIS est donc majoritairement froid, sous une forme dite atomique ou neutre; on parlera aussi de région H I. Il s'étend sur des distances très importantes le long du disque, au delà même de la composante stellaire de la galaxie où il est capable de se régénérer par accréation du milieu inter-galactique (voir par exemple van de Voort et al., 2011). On y distingue généralement deux composantes, le milieu atomique « tiède » (5000 K) d'une densité de 0.1 atome/cm³ et le milieu froid (≈ 100 K) dont la densité est de 30 atomes/cm³ (Draine, 2011). Ces deux composantes joignent à leur bout les composantes moléculaire et ionisé. En effet, proche des étoiles, le gaz soumis à leur radiation s'ionise et devient plus diffus, atteignant des températures entre 10⁴ à 10⁶ K (Draine, 2011). À l'inverse, sous certaines conditions permettant des températures de quelques dizaines de Kelvin, la molécule de dihydrogène (H₂) est capable de se former. On atteint alors la phase moléculaire du MIS. Ces régions sont aussi les plus denses, variant de 10² à 10⁶ atomes/cm³ et vont constituer le lieu propice à la formation d'étoiles (Draine, 2011).

Le MIS n'est pas constitué entièrement de gaz. 1% de sa masse se trouve sous une forme solide appelé poussière². Bien que négligeable en masse, les grains de poussière jouent un rôle déterminant dans la chimie du milieu interstellaire, comme catalyseur à la formation de la plupart des molécules des régions H₂. Ils offrent ainsi la voie privilégiée pour former le dihydrogène au sein du MIS dense, ainsi que son principal traceur : le CO. Si la formation des régions H₂ semble émerger du milieu atomique dense grâce aux poussières, l'origine de ces nuages moléculaires à des échelles de l'ordre de la dizaine à la centaine de parsecs reste encore une question débattue (voir ci-après ainsi que les revues de Klessen & Glover, 2016; Chevance et al., 2020).

La deuxième particularité de la poussière vient du fait qu'elle absorbe efficacement la lumière visible des étoiles. Cette propriété est une limite importante à l'observation des régions moléculaires qui apparaissent

2. Ces grains de poussière sont bien plus petits que ce que la connotation du mot laisse envisager. La taille de ces grains varient du nano- au micro-mètre, correspondant plus à de la fumée de cigarette.

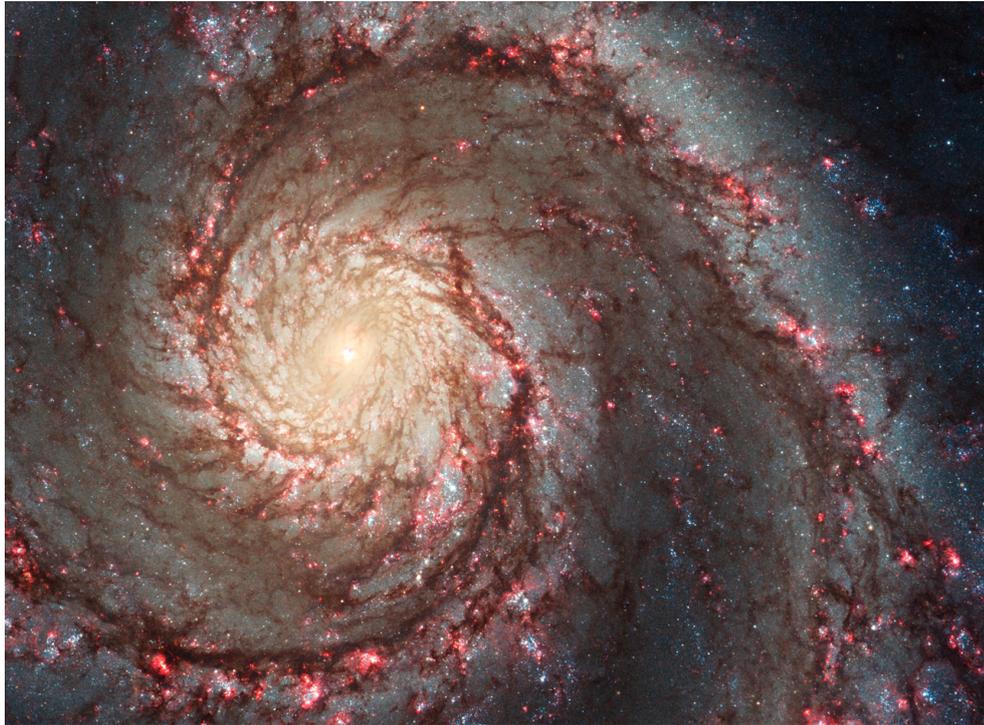


FIGURE 1.1 – Composition colorée de la galaxie M51. La dynamique de cette galaxie crée des ondes de densités lui donnant son aspect spiral. Les régions les plus denses contiennent aussi le plus de poussières ce qui rend ces régions les plus obscures. Ces régions sont le lieu de la formation d'étoiles. On distingue émergeant de ces régions des bulles rouges lumineuses indiquant la présence d'hydrogène ionisé. C'est la rétroaction du rayonnement des étoiles jeunes sur le gaz environnant. Enfin on remarque aussi la présence d'étoiles bleues en amas sur l'extérieur des bras spiraux. Ces étoiles ont « nettoyé » leur environnement de gaz et s'échappent des bras spiraux pour peupler le disque de nouvelles étoiles. Crédits : NASA, ESA, S. Beckwith (STScI) et la Hubble Heritage Team (STScI/AURA).

ainsi comme des nuages sombres. C'est seulement depuis l'avènement de l'astronomie radio et infrarouge que leur structure interne a été révélée. La Fig.1.1 montre une vue de la galaxie M51, une galaxie semblable à la Voie Lactée, en lumière visible et infrarouge proche. Les régions rouges sombres correspondent aux régions obscurcies par la poussière indiquant une densité du MIS plus importante. Les quelques zones lumineuses rosées indiquent les régions de formations stellaires où les étoiles récemment formées ionisent le gaz environnant créant ces bulles caractéristiques.

1.2 Processus physiques de la formation d'étoiles

1.2.1 Effondrement et fragmentation

L'observation des galaxies indique que les nuages moléculaires sont abondants au sein des disques des galaxies spirales. Comme on peut le voir sur la Fig.1.1, ils se situent le long des bras spiraux. Dans notre galaxie, ils se concentrent pour la plupart au sein d'un anneau qui se situe entre 3 et 8 kpc du centre galactique (Draine, 2011). Ces derniers s'étendent sur des échelles allant du parsec à la centaine de parsecs où l'on parlera alors

de nuage moléculaire géant ou GMC (Rice et al., 2016; Miville-Deschênes et al., 2017). La masse importante de gaz au sein de ces GMC implique que la gravitation constitue un des principaux moteurs conduisant à la concentration du gaz pour former les étoiles. Dans le cas d'un nuage sphérique et de densité uniforme, le temps caractéristique d'un effondrement sans force opposée – dénommée temps de chute libre – vaut :

$$t_{\text{ff}} = \sqrt{\frac{3\pi}{32G\rho_0}}, \quad (1.1)$$

où G est la constante gravitationnelle et ρ_0 la densité moyenne du nuage. La conséquence évidente de la dépendance en densité implique que plus un nuage sera dense, plus ce temps sera court et ainsi l'effondrement efficace. De plus, comme ce dernier conduit à l'augmentation en densité du nuage, le phénomène est sujet à un effet d'emballlement. Il existe en revanche, de nombreux processus à l'œuvre au sein du MIS à même de s'opposer à cet effondrement. Le plus simple de ces processus est l'agitation thermique du gaz qui agit comme une pression de l'intérieur vers l'extérieur. La théorie cinétique des gaz permet de définir la température comme une mesure de l'énergie cinétique de ces particules. Dans le cas d'un gaz parfait monoatomique :

$$E_c = \frac{3}{2}Nk_B T, \quad (1.2)$$

où N est le nombre de particules, T la température et k_B la constante de Boltzmann. En ne considérant que l'agitation thermique du gaz et l'auto-gravitation, le théorème du viriel implique qu'un tel système est à l'équilibre si :

$$2E_c + E_p = 0, \quad (1.3)$$

avec $E_p = -\frac{3}{5}\frac{GM^2}{R}$ pour une sphère uniforme de gaz. On peut ainsi déduire la masse d'un nuage à l'équilibre pour une densité et une température donnée.

$$M_J = \left(\frac{5k_B T}{G\mu m_H}\right)^{\frac{3}{2}} \left(\frac{3}{4\pi n_H}\right)^{\frac{1}{2}}, \quad (1.4)$$

où n_H , est la densité particulaire de l'hydrogène moléculaire, m_H est la masse de l'hydrogène et $\mu = 2.33$ est le poids particulaire moyen pour le milieu moléculaire. Cette masse représente la masse critique à partir de laquelle un nuage s'effondre sous son propre poids et est noté M_J pour masse de Jeans. La dépendance en température et en densité de ce critère montre qu'un nuage sera plus à même de s'effondrer s'il est dense et froid, ce qui correspond aux conditions que l'on rencontre dans la phase moléculaire du MIS. Pour des conditions denses du milieu moléculaire où $T \sim 10\text{K}$ et $n_H = 10^6$ atomes/cm³, la masse de Jeans est de l'ordre de la masse du soleil. Or la masse typique d'un GMC est de l'ordre du $10^6 M_\odot$ (Draine, 2011) qui n'est donc pas stable thermodynamiquement. De plus, un tel nuage n'est jamais de densité uniforme, et les fluctuations de densité en son sein implique que plusieurs surdensités atteignant la masse de Jeans peuvent s'effondrer indépendamment les unes des autres. Ces effondrements conduisent mécaniquement à une augmentation de la densité du gaz, diminuant encore la masse de Jeans et conduisant à une amplification du processus. Cette description permet d'appréhender de façon heuristique cette réaction en cascade que l'on appelle la fragmentation. L'observation des nuages moléculaires montre la prééminence de cette structure fragmentée sur une large plage de distance

allant de la taille des bras spiraux (comme en témoigne la carte des nuages moléculaires du disque de la Voie Lactée par Dame et al., 2001) jusqu'à de petites régions de quelques fractions de parsecs, comme l'a dévoilé le télescope spatial *Herschel* (André et al., 2010; Arzoumanian et al., 2013; Molinari et al., 2010). La Fig.1.2 montre un exemple de la structure hiérarchique d'une région de formation d'étoiles s'étendant sur quatre ordres de grandeurs.

Néanmoins l'observation de ces structures fragmentées à différentes échelles impliquent de nombreux autres processus complexifiant le raisonnement développé plus haut. Les lois dites de Larson (1981), ont très tôt indiquées que le principal agent contre-balançant les effets de la gravité, n'était pas la pression thermique mais le mouvement turbulent du gaz, injectant à toute échelle de l'énergie, et pouvant empêcher le gaz de se condenser. Cependant, la turbulence crée aussi un réseau complexe d'ondes de chocs dont les nœuds sont les surdensités du nuage propices à l'effondrement. Ce double rôle du mouvement turbulent du MIS suscite encore de nombreux débats quant à son impact précis sur la formation d'étoiles (voir p. ex. Mac Low & Klessen, 2004; Ballesteros-Paredes et al., 2007; Klessen, 2011). De plus, son origine semble être multiple regroupant de nombreux processus comme les mouvements à grandes échelles du gaz dus à la rotation galactique ou alimentés par la rétroaction violente des étoiles. De ce fait, il représente aussi un ingrédient important dans le but d'appréhender l'enchevêtrement des différentes phases du MIS et la constitution des nuages moléculaires (voir p. ex. les revues de Klessen & Glover, 2016; Chevance et al., 2020).

1.2.2 Formation d'une proto-étoile

L'effondrement d'un nuage moléculaire entre dans une deuxième phase lorsque la fragmentation s'arrête. En effet, la contraction du gaz lui transmet de l'énergie. Ce dernier est capable de se refroidir par les raies de structure fine du gaz et le rayonnement infrarouge de la poussière, ce qui permet par ailleurs de l'observer dans l'infrarouge lointain. La phase est alors isotherme et la fragmentation se poursuit. Néanmoins lorsque le refroidissement n'est plus assez efficace, le gaz se met à chauffer. Il devient optiquement épais. Sous ses conditions l'équation d'état du gaz est modifiée et la masse de Jeans augmente, de sorte qu'un cœur en quasi-équilibre hydrostatique se forme (on parle aussi du premier cœur de Larson). Le nuage ne se fragmente plus et l'on parle alors de limite d'opacité pour la fragmentation (Bate et al., 2003). Ce changement de régime dépend de l'opacité du gaz et de sa température initiale (Masunaga et al., 1998; Vaytet et al., 2012) et intervient typiquement à une densité de $\sim 10^{-13}$ g/cm³. Ce changement de régime n'arrête néanmoins pas la contraction qui se poursuit mais de façon adiabatique cette fois-ci. La température au centre du nuage grimpe ainsi très rapidement passant de 10 K à 2000 K, température à laquelle la molécule de dihydrogène se dissocie. Enfin, une fois qu'une grande partie du dihydrogène est dissociée, le centre du nuage, approchant une densité de $\sim 10^{-3}$ g/cm³, devient le lieu de formation d'un second cœur en équilibre hydrostatique (ou second cœur de Larson). Ce dernier constitue l'embryon d'étoile en train de se former.

Cette étape constitue la première phase pré-stellaire et est difficile à observer, non seulement car l'enveloppe de gaz qui l'alimente représente encore la majorité de la masse et obscurcit le cœur du nuage, mais aussi car elle est très courte de l'ordre de 10 000 années (on parle alors d'objet de classe 0; André et al., 2000). Rapidement le gaz continuant d'être accrété par l'objet central s'aplatit en un disque par conservation du moment cinétique, et forme la deuxième phase proto-stellaire. L'activité magnétique s'intensifie et crée des jets bi-polaires

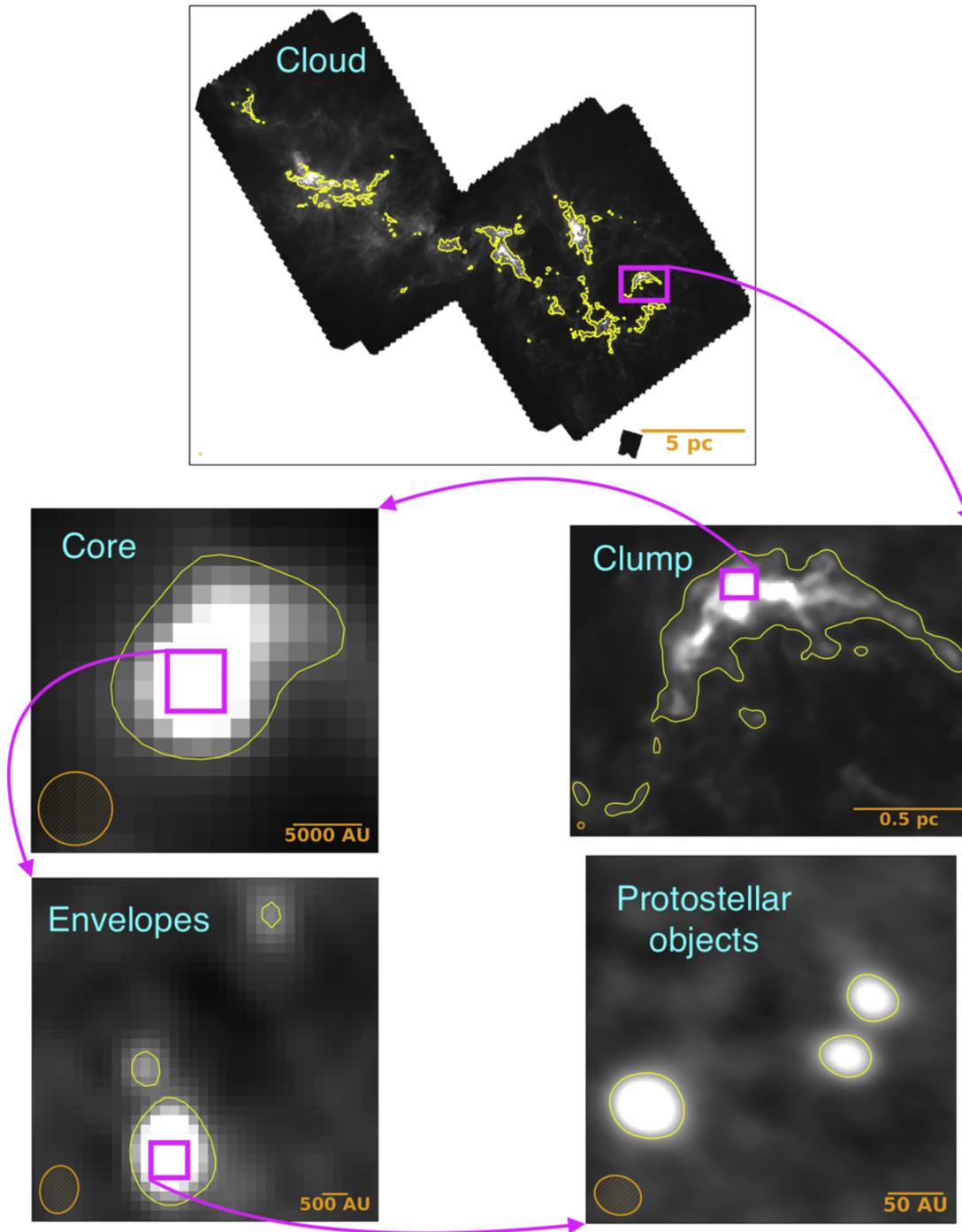


FIGURE 1.2 – Observations à différentes échelles de la région du nuage de Persée. Chaque panneau montre une région du nuage à un ordre de grandeur inférieur. Différentes observations sont nécessaires pour réaliser cette composition. Cela illustre la complexité de l'exercice ainsi que de son interprétation. Les panneaux *Cloud* et *Clump* sont issus de l'observation à $350 \mu\text{m}$ du télescope spatial *Herschel* (Zari et al., 2016); *Core* de l'observation à $850 \mu\text{m}$ de SCUBA (Di Francesco et al., 2008); *Envelopes* à 1.3 mm de l'interféromètre SMA; et *protostellar objects* (objets proto-stellaires) de l'observation combinée à 8 mm et 1 cm de l'interféromètre VLA (Tobin et al., 2016). Image extraite de Pokhrel et al. (2018).

dispersant l'enveloppe au niveau des pôles (objet de classe I). Les processus d'accrétion se poursuivent jusqu'à épuisement de l'enveloppe de gaz dans laquelle la proto-étoile était immergée. La majorité de la masse est ainsi dans l'objet central qui est maintenant considéré comme une étoile de la pré-séquence principale (objet de classe II). Le disque proto-stellaire subsiste néanmoins pour devenir peu à peu un disque proto-planétaire. La contraction de la jeune étoile se poursuit faisant encore augmenter la température en son cœur permettant de commencer les premières réactions de fusions thermonucléaires (objet de classe III). Lorsque la fusion de l'hydrogène commence à 10^7 K, l'étoile commence officiellement sa vie sur la séquence principale.

Ces dernières étapes contrairement aux premières sont beaucoup plus lentes. Elles dépendent de la masse qui a réussi à être accrétée; pour une étoile de $1 M_{\odot}$ cette phase dure ~ 30 millions d'années (M.a.), alors que pour une étoile de $0.5 M_{\odot}$, elle est de l'ordre de ~ 100 M.a. Il existe même une limite en dessous de laquelle, l'étoile n'atteindra jamais la fusion de l'hydrogène. Cela concerne les étoiles de moins $0.08 M_{\odot}$. Ce sont les naines brunes. Ces dernières sont quand même considérées comme des étoiles puisque la fusion d'autres éléments comme le deutérium y est possible.

À l'inverse pour les étoiles massives, cette séquence est fortement remise en cause. En effet, les étoiles ayant une masse supérieure à $8 M_{\odot}$, atteignent la séquence principale en moins d'un million d'années, alors qu'elles sont encore immergées dans leur enveloppe natale (Motte et al., 2018). De plus, leur forte pression de radiation est à même de stopper toute accrétion supplémentaire (voir p. ex. Kahn, 1974, qui avait très tôt pointé cette difficulté pour des étoiles autour de $40 M_{\odot}$). Néanmoins, cette séquence d'évolution développée initialement dans Larson (1969) et Shu et al. (1987) permet d'appréhender les principaux processus permettant de transformer le gaz en étoile. Pour les étoiles de faibles masses, les récentes simulations à très haute résolution incluant les champs magnétiques et un traitement poussé du transfert radiatif ont permis d'affiner notre compréhension de ces différentes étapes sans remettre le schéma expliqué ci-dessus en cause (Vaytet et al., 2012, 2013; Hennebelle et al., 2017). Ainsi, on distingue aujourd'hui la formation d'étoiles de faible masse de celle des étoiles massives (voir respectivement les revues de Andre et al., 2000; Motte et al., 2018). Pour ces dernières, notre compréhension reste incomplète. Si d'important niveaux de turbulence et/ou de champs magnétiques pourraient permettre l'accrétion du gaz à la manière du schéma classique expliqué ci-dessus (McKee & Tan, 2003), d'autres théories privilégient d'autres canaux permettant d'accumuler une spectaculaire quantité de matière en un seul objet. Certains scénarios suggèrent par exemple une accrétion partagée et « compétitive » entre plusieurs proto-étoiles au sein d'une même enveloppe (Bonnell et al., 2001, 2007; Vázquez-Semadeni et al., 2019), tissant un lien plus complexe entre les propriétés des nuages et les étoiles qu'ils engendrent. Ces théories ont permis d'appréhender la formation des étoiles en lien plus étroit avec leur environnement. Dans ce contexte, la limite de résolution des observations ainsi que des simulations est encore plus critique et laisse encore des questions fondamentales en suspens, comme l'universalité de la fonction de masse initiale des étoiles (IMF), la distribution spatiale de la masse des étoiles à leur naissance (ségrégués ou non) ou encore la quantité globale de gaz transformée en étoiles (voir p. ex. Krumholz, 2014).

1.2.3 Fin de la formation d'étoiles

L'effondrement d'un nuage en plusieurs fragments puis en étoiles indique que ces dernières ne se forment pas seules. De plus, l'effet d'emballement de l'effondrement du gaz conduit à l'émergence des jeunes étoiles

quasi simultanément. Les simulations de régions de formation d'étoiles s'accordent sur le fait que ce processus est rapide, se déroulant sur un temps proche du temps de chute libre (voir p. ex. [Krumholz et al., 2012](#); [Grudić et al., 2018](#)). Les derniers processus d'accrétion puis l'ignition du rayonnement des jeunes étoiles ont pour effet d'épuiser le réservoir de gaz du nuage. La rétroaction du rayonnement sur le nuage environnant suscite un intérêt particulier car il est considéré comme la principale limitation à la masse maximale des étoiles mais aussi à la formation d'autres étoiles voisines. La prise en compte précise de ces effets est néanmoins délicate. Les différents types de rayonnement capable d'ioniser le gaz sont nombreux et dépendent des sources considérées. Si les étoiles de faible masse n'émettent que peu de rayonnement ultraviolet (UV), les jets bipolaires qu'elles créent pendant les phases proto-stellaires produisent des rayons X créant des chocs ([Bally, 2016](#)). En revanche plus les étoiles sont massives, plus leur rayonnement ionisant sera important et efficace pour disperser le gaz ([Matzner, 2002](#)).

L'observation de régions de formation stellaire à différents stades d'évolution permet de mesurer l'efficacité du processus de formation d'étoiles ϵ au cours du temps t , c.-à-d. quelle fraction de masse du gaz est transférée aux étoiles.

$$\epsilon = \frac{M_{\text{étoiles}}}{M_{\text{étoiles}} + M_{\text{gaz}}} \quad (1.5)$$

avec $M_{\text{étoiles}}$ la masse totale des étoiles et M_{gaz} la masse de gaz résiduelle. L'efficacité mesurée est en fait $\epsilon_{\text{ff}} = \epsilon/t_{\text{ff}}$ qui est la mesure au moment des observations et rapportée au temps de chute libre théorique du nuage considérée. On l'estime généralement en comptant le nombre de jeunes étoiles dans un volume donné d'une région dont on estime la masse en gaz et dont le temps d'effondrement théorique t_{ff} n'est fonction que de la densité (voir équation 1.1). Ainsi $\epsilon_{\text{ff}} = 1$ équivaut à une conversion totale d'un nuage de gaz en étoile, durant un effondrement purement gravitationnelle. Les études observationnelles s'accordent généralement sur une valeur de l'ordre du pourcent (voir la revue de [Krumholz et al., 2019](#)). Cette valeur constitue une contrainte forte pour les simulations qui tentent de démêler la contribution des nombreux processus limitant cet effondrement, comme la rétroaction du rayonnement des étoiles, le niveau de turbulence du MIS ou le rôle des champs magnétiques. À ce jour, l'idée que le processus de formation d'étoiles est largement inefficace par nature, reste encore à élucider.

1.3 Amas d'étoiles

La définition classique d'un amas d'étoiles consiste en une structure stellaire liée par sa propre gravité, c.-à-d. que son énergie totale est négative soit $E_{\text{tot}} = E_c + E_p < 0$. Cette définition minimaliste doit être adaptée dans le cas de régions de formation d'étoiles car le gaz qui n'a pas été converti en étoiles peut constituer une part importante de la masse totale de la région et donc contribuer significativement au potentiel gravitationnel. Une autre définition basée sur des mesures observationnelles est souvent adoptée, p. ex. [Lada & Lada \(2003\)](#) considère tout regroupement de plus de 35 étoiles et dont la densité stellaire dépasse $1 M_{\odot}/\text{pc}^3$ comme un amas. Leur étude montre notamment que le nombre de ce type d'amas au sein de la Voie Lactée était anti-corrélé à leur âge. Ils estimaient ainsi que 90% des amas jeunes encore immergés se dissolvaient en moins de 10 M.a. dans le disque galactique. La dispersion rapide du gaz résiduel en quelques millions d'années ([Krause et al., 2016](#)) fût rapidement désignée comme la cause principale. L'efficacité globale du processus de formation

devait alors être supérieur à 30% pour qu'un amas subsiste lié (Boily & Kroupa, 2003a,b). Cette efficacité largement supérieur aux quelques pourcents généralement observés, pose clairement la question de la possibilité de formation d'amas durablement liés (Krumholz & McKee, 2020).

Néanmoins les étoiles formées en petits groupes héritent de la structure fragmentée du gaz dont ils sont issus. Cela implique qu'ils ne sont initialement pas à l'équilibre et subissent dès les premiers instants de leur formation, de l'évolution dynamique. La dynamique à N -corps indique qu'un système d'étoiles en auto-gravitation évolue selon un processus de diffusion engendré par les rencontres successives que subit chaque étoile avec ses voisines. On parle alors de diffusion à deux corps ou plus simplement de relaxation. Il correspond au temps nécessaire pour que les interactions successives des étoiles entre elles, conduisent à modifier de manière significative leurs orbites. Cet effet est local et dépend de la granularité et de la densité du système. Pour un système à l'équilibre et sphérique, on l'approxime alors par le temps de relaxation à mi-masse (Meylan & Heggie, 1997) :

$$t_{\text{rh}} = \frac{0.138}{\ln 0.4N} \sqrt{\frac{N}{G\langle m \rangle}} R_{\text{h}}^{\frac{3}{2}}, \quad (1.6)$$

où R_{h} est le rayon à mi-masse du système, N le nombre de particule du système et $\langle m \rangle$ la masse moyenne des étoiles du système. Dans le cas de jeunes amas contenant quelques centaines d'étoiles s'étendant sur quelques dixièmes de parsecs, ce temps est de l'ordre de quelques millions d'années. Il donne une borne supérieure dans le cas d'amas fragmentés à l'effacement des conditions initiales. Cet effet est particulièrement marqué dans les amas les plus denses comme celui de la nébuleuse d'Orion (ONC). Cet amas bien qu'encore enfoui dans du gaz et dont l'âge est estimé autour de 3 M.a. apparaît avoir subi assez d'évolution dynamique pour arborer une structure relativement sphérique et relaxée. Lorsque ONC a commencé à former des étoiles, cet agencement était probablement bien différent.

Comprendre ces effets dynamiques permet non seulement de remonter aux premiers instants de la formation d'étoiles mais aussi de prédire le futur de ces régions. En effet, le processus de relaxation qui implique une redistribution d'énergie au sein des étoiles jeunes peut altérer significativement le nombre d'étoiles se dissolvant dans le champ même une fois le gaz expulsé (Farias et al., 2015). L'évolution dynamique à long terme de ces régions permet ainsi de dessiner une séquence d'évolution vers des amas plus évolués comme les amas ouverts voire globulaires. La dissolution d'une partie des étoiles jeunes brouille les limites entre amas d'étoiles liés et étoiles du champ (on parle généralement d'associations pour désigner ces structures jeunes en dissolution). Les amas ouverts souvent considérés comme liés, représentent ainsi souvent les surdensités de structures plus larges, généralement non liées entre elles, sur des échelles pouvant atteindre plusieurs kiloparsecs (Gouliermis, 2018). À l'ère de *Gaia* DR2 (Gaia Collaboration et al., 2018) et maintenant DR3 (Gaia Collaboration et al., 2020), une vision globale liant à grande échelle des structures naguère distinctes, se dessine (voir notamment les catalogues de Kounkel & Covey, 2019; Cantat-Gaudin et al., 2020).

1.4 Approche numérique de la formation d'étoiles

Les observations des régions de formations d'étoiles étant limitées (absorption de la lumière visible par la poussière et rareté relative de ces régions dans le voisinage solaire), différents modèles théoriques générale-

ment basé sur des simulations numériques ont été nécessaire à l'établissement de notre connaissance des processus de formation des étoiles. La puissance de calcul grandissante des ordinateurs a permis depuis 50 ans de faire des avancées considérables, permettant de contraindre de plus en plus de détails. Néanmoins les échelles impliquées par ce processus obligent les théoriciens à simplifier le problème. Ainsi les premiers calculs de [Larson \(1969\)](#) et [Shu \(1977\)](#) consistait à contraindre l'effondrement d'un nuage de gaz en une étoile seule via une approche uni-dimensionnelle. Ces premiers travaux ont posé les bases de l'effondrement en deux cœurs des nuage pré-stellaires. Les études suivantes se sont intéressées au phénomène de fragmentation des nuages. Ce processus est asymétrique et nécessite donc la création de simulations à trois dimensions. Au fil de l'amélioration de leur résolution, le processus d'accrétion des proto-étoiles par l'intermédiaire d'un disque circumstellaire a pu y être progressivement intégré. Les simulations de [Bate et al. \(2003\)](#) furent pionnières car elles permirent de capturer de manière globale l'effondrement isotherme d'un nuage de ~ 0.4 pc jusqu'à la formation du premier cœur hydrostatique de Larson ~ 5 UA. Ces simulations, ainsi que celles qui ont suivies (p. ex. [Bonnell et al., 2008](#); [Bate, 2009, 2012](#)) ont permis de mettre en lumière la formation en petit groupes de quelques dizaines à centaines d'étoiles. Néanmoins la puissance de calcul pour réaliser de telles simulations est généralement gigantesque. Par exemple, la simulation de [Bate \(2012\)](#) la plus résolue ~ 1 UA similaire à celle de [Bate et al. \(2003\)](#) a nécessité 6.4 millions d'heures CPU. Ainsi, une mise à l'échelle dans le but de modéliser la fragmentation d'un nuage de plusieurs dizaines de parsecs ne peut se faire qu'au prix d'une résolution limitée; quoique la résolution des calculs numériques soit toujours en progression (voir p. ex. [Wall et al., 2019, 2020](#))

De plus, ces simulations à haute résolution ne traitent généralement que les effets hydrodynamiques. Certains processus de chauffage du gaz sont éventuellement pris en compte dans le but de limiter la sur-fragmentation, notamment au sein des disques circumstellaires ([Bate, 2012](#)). Néanmoins la prise en compte détaillée des processus radiatifs de rétroaction des étoiles, de refroidissement du gaz, de vents stellaires, ainsi que le couplage avec les champs magnétiques est encore limitée. L'émergence de nombreux codes permettant d'intégrer ces différents processus physiques a permis aux simulations des dernières années de traiter le problème via une approche multiphysique. Ainsi l'impact des différents processus de la rétroaction du rayonnement a pu être évalué de manière détaillée sur les disques circumstellaires ([Klassen et al., 2016](#)) mais aussi à l'échelle de régions entières ([Grudić et al., 2018](#)). Les derniers travaux de [Bate \(2019\)](#) s'intéressent à l'influence des métaux dans les équilibres thermo-chimiques du gaz. Le rôle des champs magnétiques est lui aussi de plus en plus étudié au sein de la formation des proto-étoiles ([Hennebelle et al., 2017](#)) comme au niveau du MIS turbulent ([Hennebelle, 2018](#)).

Néanmoins, comme de nombreux problèmes en astrophysique la formation d'étoiles reste un problème aux conditions initiales. Ainsi le nombre restreint de calculs à haute résolution ne permet pas de sonder l'ensemble de l'espace des paramètres. Ainsi une approche complémentaire à ces calculs consiste encore à simplifier le problème dans le but de générer de nombreuses configurations et les analyser de manière statistique. C'est aussi une méthode utilisée pour appréhender la formation des étoiles à des échelles supérieures.

Ce sera d'ailleurs l'approche utilisée dans ce travail de thèse. On se basera notamment sur les travaux de [Dorval et al. \(2016\)](#); [Dorval et al. \(2017\)](#) dans le but de modéliser la formation de structures stellaires représentatives des régions de formation d'étoiles en utilisant une approche purement aux N -corps. Ceci nous permettra d'étudier la dynamique complexe des étoiles aux premiers stades de leur vie. Il sera aussi possible d'étudier

des ensembles représentatifs d'amas jeunes massifs s'étendant sur quelques parsecs et comportant plusieurs milliers d'étoiles.

1.5 La plateforme AMUSE

L'engouement pour l'approche multiphysique a conduit à complexifier de manière surprenante les codes de simulation astrophysique. Au développement de codes hier dédiés à la résolution d'une seule physique, s'ajoute différentes extensions dans le but d'intégrer le traitement d'autres processus. L'exemple le plus emblématique est peut-être le code RAMSES (Teyssier, 2002) qui était à l'origine un simple code hydrodynamique et qui aujourd'hui est capable d'intégrer des processus de transfert de rayonnement, de magnéto-hydrodynamique (MHD), de chimie du MIS, etc.. Dans le domaine de la dynamique N -corps, la suite NBODY (Aarseth, 1999) fût pionnière et a été déclinée pour traiter l'évolution stellaire, divers effets de marées (Renaud et al., 2011) ou encore intégrer des aspects de relativité générale (Aarseth, 2012). Les phases de développement et de test d'un nouveau code sont en effet longues et coûteuses. Il est ainsi plus efficace d'étendre les possibilités d'un code déjà existant que de repartir de zéro. En revanche, ces ajouts ne peuvent pas radicalement bouleverser la structure algorithmique du code original au risque d'entraver son fonctionnement naturel. Outre le fait de limiter les possibilités dans l'ajout de nouvelles physiques, cela crée généralement de vraies « usines à gaz » dont la prise en main devient véritablement compliquée.

À rebours de cette conception, une autre initiative s'est développée, au milieu des années 2000, avec le projet AMUSE pour *Astrophysical Multi-Unit Software Environment*. AMUSE est le résultat du développement d'un logiciel plus ancien dénommé StarLab³ (Hut et al., 2010; Portegies Zwart et al., 1999). Ce dernier avait réussi à associer plusieurs codes de dynamique gravitationnelle et d'évolution stellaire au sein d'une interface dont la structure ne dépendait d'aucun d'entre eux. Néanmoins les choix algorithmiques réalisés ne permettaient pas d'étendre plus loin ses possibilités pour ajouter de l'hydrodynamique par exemple. C'est ainsi qu'AMUSE vit le jour et son mode de fonctionnement fût pensé avec un réel souci de modularité pour ainsi étendre ses capacités à l'infini.

La philosophie de cette plateforme logicielle repose sur le fait qu'il n'y a pas de code maître pour diriger la simulation. Ainsi l'utilisateur est libre de modifier la structure algorithmique de la simulation à souhait. De plus, AMUSE intègre au sein d'une même interface, des codes issus de la communauté astrophysique, dans le but de faciliter leur prise en main et de pouvoir les coupler entre eux. Aujourd'hui, c'est plus de 50 codes qui peuvent être utilisés d'une manière facilitée grâce à la plateforme. Ces codes intègrent des domaines de la dynamique gravitationnelle et l'évolution stellaire mais aussi la (magnéto-)hydrodynamique et le transfert du rayonnement. La plateforme peut être librement téléchargé via le dépôt officiel⁴. Il est toujours en développement actif et de nouveaux codes peuvent y être intégrés par la communauté. La modularité de la plateforme lui confère des possibilités importantes de développement⁵. Le livre *Astrophysical Recipes* (Portegies Zwart &

3. <http://www.sns.ias.edu/~starlab/starlab.html>

4. <https://github.com/amusecode/amuse>

5. Il existe d'ailleurs une version minimaliste de la plateforme intégrant juste le squelette d'AMUSE, nommé minimUSE : <https://github.com/amusecode/minimuse>, qui permet d'étendre l'usage de la plateforme à d'autres domaines que l'astrophysique, comme l'océanographie : <https://github.com/omuse-geoscience/omuse> ou l'hydrologie : <https://github.com/eWaterCycle/HyMUSE>.

McMillan, 2018) écrit par les fondateurs constitue une documentation substantielle et offre une vision globale du fonctionnement et de ses possibilités. De nombreuses situations concrètes y sont aussi discutées. Portegies Zwart et al. (2013) offre une description plus ancienne mais plus succincte ainsi que Pelupessy et al. (2013) détaillent quelques tests de base réalisés sur les codes intégrés à AMUSE.

Du fait de son fonctionnement, AMUSE a été beaucoup utilisé dans le but de traiter la nature multiphysique des régions de formation d'étoiles (voir entre autres : Pelupessy et al., 2004; Fujii et al., 2012; Fujii & Portegies Zwart, 2015, 2016; Sills et al., 2018; Kamdar et al., 2019; Wall et al., 2019, 2020). C'est aussi le choix qui a été fait dans le cadre de cette thèse.

1.6 Plan de thèse

Ce travail de thèse s'intéressera donc aux processus dynamiques à N -corps au sein des régions de formations d'étoiles à l'aide de simulations numériques. Nous nous focaliserons sur l'apport de ces calculs théoriques dans le but de mieux interpréter les observations.

Si l'idée que les étoiles se forment majoritairement en groupes est bien établi, ces derniers évoluent rapidement, alors même qu'ils sont encore enfouis dans du gaz résiduel. L'extinction due à la poussière qui y est immergée pose de sérieuses limites à l'observation des premiers stades de la conversion du gaz en étoiles. Les propriétés dynamiques de ses groupes primordiaux sont donc difficilement contraintes et seulement pour un ensemble limité de régions (Joncour et al., 2017; Gaczkowski et al., 2013; Hacar et al., 2017). L'apport de calcul théorique reproduisant de manière réaliste l'état initial des structures stellaires dans l'espace des phases est donc crucial pour étudier les processus à N -corps régissant leur évolution. Préciser l'impact de ces effets dynamiques permet de mieux identifier les propriétés issues soit du processus de formation d'étoiles, ou de leur simple évolution dynamique. Dans le cas de la formation d'étoiles massives, cela peut permettre d'apporter les indices visant à étayer l'un des nombreux scénarios concurrents. Cela permettra d'aborder des questions comme la correspondance entre la dispersion des vitesses des cœurs pré-stellaires et des jeunes étoiles (Foster et al., 2015), ou de leur stabilité dynamique à différents stades dans une perspectives d'évolution à long terme.

En raison de la nature collisionnel de ces systèmes, cette évolution chaotique nécessite une intégration précise des équations du mouvement par des codes spécifiques. Le premier chapitre introduira les codes de calculs N -corps intégrés à la plateforme AMUSE et détaillera quelques uns des tests réalisés pour évaluer leur performance. Ces tests permettront de dégager un code qui sera utilisé dans le reste des travaux.

Le deuxième chapitre rappellera les principes fondamentaux de la méthode de fragmentation à N -corps (GDF) développée par Dorval et al. (2017), permettant de reproduire des distributions stellaires, représentatives d'amas jeunes. Deux méthodes d'identifications de groupes, le Minimum Spanning Tree (MST) et l'algorithme HOP, seront testés sur ces configurations dans le but d'extraire les surdensités stellaires considérés comme les graines de la formation d'étoiles.

Dans le troisième chapitre, nous prendrons une posture d'observateurs. Ainsi, nous identifierons les groupes sur-denses au sein de nos modèles vus en projection avec la méthode du MST. Nous quantifierons les biais de projection des mesures de leurs propriétés dynamiques comme leur taille et leur dispersion des vitesses. Nous montrerons en particulier que le calcul de leur masse dynamique est une fonction forte de la contamination

des étoiles proches, mais non liés aux groupes sur la ligne de visée.

Le quatrième chapitre revient plus en détail sur les propriétés mesurées des groupes identifiés par le MST sans contamination. Ces propriétés dynamiques fondamentales permettent de mieux comprendre l'histoire de formation de ces groupes. Nous rappellerons entre autre que la ségrégation en masse des étoiles au sein des modèles GDF, et qui est aussi observée dès le début de la vie de jeunes amas (Plunkett et al., 2018) est un des aspects majeurs à prendre en compte dans le but de contraindre le rapport masse/lumière des groupes. De plus, nous tissons un lien entre la dynamique de nos groupes avec un ensemble d'amas et de sous-amas identifiés grâce au catalogue DR2 de la mission *Gaia* (Kuhn et al., 2019). Cela justifiera la force de la méthode GDF dans le but d'extraire des tendances statistiques concernant les propriétés de ces groupes.

Le cinquième chapitre s'intéressera à de plus petites structures stellaires, à savoir les systèmes binaires et multiples. Ces systèmes connus pour interagir violemment au sein d'environnements denses, modifient les propriétés dynamiques des amas jeunes. Nous reviendrons tout d'abord sur la méthode choisie pour leur prise en compte précise dans nos calculs, grâce à AMUSE. Puis, nous comparerons les populations de binaires créées dynamiquement au sein des modèles GDF avec les populations observées dans le champ.

Enfin le dernier et sixième chapitre conclura ce travail par de potentiels voies futures de recherche. En intégrant une composante gazeuse à nos modèles, il serait possible d'étudier plus précisément la dynamique des étoiles au sein du nuage où elles naissent. Le cas d'un système binaire dans un environnement gazeux est envisagé. De plus, nous insistons sur l'importance de la prise en compte de l'extinction au sein de ces régions pour lever d'autres biais observationnels. Les modèles GDF couplés à des cartes d'extinction 3D pourraient former une base solide dans le but de créer des observations de synthèse. Enfin l'intégration des modèles GDF à un environnement galactique à plus grande échelle, permise par la modularité de la plateforme AMUSE, permet d'envisager l'évolution des jeunes amas dans un contexte plus global.

BIBLIOGRAPHIE

- Aarseth, S. J. 1999, *PASP*, **111**, 1333
- Aarseth, S. J. 2012, *MNRAS*, **422**, 841
- André, P., Men'shchikov, A., Bontemps, S., et al. 2010, *A&A*, **518**, L102
- Andre, P., Ward-Thompson, D., & Barsony, M. 2000, in *Protostars and Planets IV*, ed. V. Mannings, A. P. Boss, & S. S. Russell, 59
- Arzoumanian, D., André, P., Peretto, N., & Könyves, V. 2013, *A&A*, **553**, A119
- Ballesteros-Paredes, J., Klessen, R. S., Mac Low, M. M., & Vazquez-Semadeni, E. 2007, in *Protostars and Planets V*, ed. B. Reipurth, D. Jewitt, & K. Keil, 63
- Bally, J. 2016, *ARA&A*, **54**, 491
- Bate, M. R. 2009, *MNRAS*, **392**, 590
- Bate, M. R. 2012, *MNRAS*, **419**, 3115
- Bate, M. R. 2019, *MNRAS*, **484**, 2341
- Bate, M. R., Bonnell, I. A., & Bromm, V. 2003, *MNRAS*, **339**, 577
- Boily, C. M. & Kroupa, P. 2003a, *MNRAS*, **338**, 665
- Boily, C. M. & Kroupa, P. 2003b, *MNRAS*, **338**, 673
- Bonnell, I. A., Bate, M. R., Clarke, C. J., & Pringle, J. E. 2001, *MNRAS*, **323**, 785
- Bonnell, I. A., Clark, P., & Bate, M. R. 2008, *MNRAS*, **389**, 1556
- Bonnell, I. A., Larson, R. B., & Zinnecker, H. 2007, in *Protostars and Planets V*, ed. B. Reipurth, D. Jewitt, & K. Keil, 149
- Bournaud, F. & Combes, F. 2002, *A&A*, **392**, 83
- Cantat-Gaudin, T., Anders, F., Castro-Ginard, A., et al. 2020, *A&A*, **640**, A1
- Chevance, M., Kruijssen, J. M. D., Vazquez-Semadeni, E., et al. 2020, *Space Sci. Rev.*, **216**, 50
- Combes, F. 2004, in *Penetrating Bars through Masks of Cosmic Dust*, ed. D. L. Block, I. Puerari, K. C. Freeman, R. Groess, & E. K. Block (Dordrecht : Springer Netherlands), 57–74

- Combes, F., Encrenaz, P. J., Lucas, R., & Weliachew, L. 1977, *A&A*, 55, 311
- Dame, T. M., Hartmann, D., & Thaddeus, P. 2001, *ApJ*, 547, 792
- Di Francesco, J., Johnstone, D., Kirk, H., MacKenzie, T., & Ledwosinska, E. 2008, *ApJS*, 175, 277
- Dorval, J., Boily, C. M., Moraux, E., Maschberger, T., & Becker, C. 2016, *MNRAS*, 459, 1213
- Dorval, J., Boily, C. M., Moraux, E., & Roos, O. 2017, *MNRAS*, 465, 2198
- Draine, B. T. 2011, *Physics of the Interstellar and Intergalactic Medium* (Princeton University Press)
- Farias, J. P., Smith, R., Fellhauer, M., et al. 2015, *MNRAS*, 450, 2451
- Foster, J. B., Cottaar, M., Covey, K. R., et al. 2015, *ApJ*, 799, 136
- Fujii, M. S. & Portegies Zwart, S. 2015, *MNRAS*, 449, 726
- Fujii, M. S. & Portegies Zwart, S. 2016, *ApJ*, 817, 4
- Fujii, M. S., Saitoh, T. R., & Portegies Zwart, S. F. 2012, *ApJ*, 753, 85
- Gaczkowski, B., Preibisch, T., Ratzka, T., et al. 2013, *A&A*, 549, A67
- Gaia Collaboration, Brown, A. G. A., Vallenari, A., et al. 2018, *A&A*, 616, A1
- Gaia Collaboration, Brown, A. G. A., Vallenari, A., et al. 2020, *arXiv e-prints*, arXiv :2012.01533
- Gouliermis, D. A. 2018, *PASP*, 130, 072001
- Grudić, M. Y., Hopkins, P. F., Faucher-Giguère, C.-A., et al. 2018, *MNRAS*, 475, 3511
- Hacar, A., Tafalla, M., & Alves, J. 2017, *A&A*, 606, A123
- Hennebelle, P. 2018, *A&A*, 611, A24
- Hennebelle, P., Lesur, G., & Fromang, S. 2017, *A&A*, 599, A86
- Hut, P., McMillan, S., Makino, J., & Portegies Zwart, S. 2010, *Starlab : A Software Environment for Collisional Stellar Dynamics*
- Joncour, I., Duchêne, G., & Moraux, E. 2017, *A&A*, 599, A14
- Kahn, F. D. 1974, *A&A*, 37, 149
- Kamdar, H., Conroy, C., Ting, Y.-S., et al. 2019, *ApJ*, 884, 173
- Klassen, M., Pudritz, R. E., Kuiper, R., Peters, T., & Banerjee, R. 2016, *ApJ*, 823, 28
- Klessen, R. S. 2011, in *EAS Publications Series, Vol. 51, EAS Publications Series*, ed. C. Charbonnel & T. Montmerle, 133–167

- Klessen, R. S. & Glover, S. C. O. 2016, *Saas-Fee Advanced Course*, 43, 85
- Kounkel, M. & Covey, K. 2019, *AJ*, 158, 122
- Krause, M. G. H., Charbonnel, C., Bastian, N., & Diehl, R. 2016, *A&A*, 587, A53
- Krumholz, M. R. 2014, *Phys. Rep.*, 539, 49
- Krumholz, M. R., Klein, R. I., & McKee, C. F. 2012, *ApJ*, 754, 71
- Krumholz, M. R. & McKee, C. F. 2020, *MNRAS*, 494, 624
- Krumholz, M. R., McKee, C. F., & Bland-Hawthorn, J. 2019, *ARA&A*, 57, 227
- Kuhn, M. A., Hillenbrand, L. A., Sills, A., Feigelson, E. D., & Getman, K. V. 2019, *ApJ*, 870, 32
- Lada, C. J. & Lada, E. A. 2003, *ARA&A*, 41, 57
- Larson, R. B. 1969, *MNRAS*, 145, 271
- Larson, R. B. 1981, *MNRAS*, 194, 809
- Mac Low, M.-M. & Klessen, R. S. 2004, *Reviews of Modern Physics*, 76, 125
- Masunaga, H., Miyama, S. M., & Inutsuka, S.-i. 1998, *ApJ*, 495, 346
- Matzner, C. D. 2002, *ApJ*, 566, 302
- McKee, C. F. & Tan, J. C. 2003, *ApJ*, 585, 850
- Meylan, G. & Heggie, D. C. 1997, *A&A Rev.*, 8, 1
- Miville-Deschênes, M.-A., Murray, N., & Lee, E. J. 2017, *ApJ*, 834, 57
- Molinari, S., Swinyard, B., Bally, J., et al. 2010, *A&A*, 518, L100
- Motte, F., Bontemps, S., & Louvet, F. 2018, *ARA&A*, 56, 41
- Pelupessy, F. I., van der Werf, P. P., & Icke, V. 2004, *A&A*, 422, 55
- Pelupessy, F. I., van Elteren, A., de Vries, N., et al. 2013, *A&A*, 557, A84
- Plunkett, A. L., Fernández-López, M., Arce, H. G., et al. 2018, *A&A*, 615, A9
- Pokhrel, R., Myers, P. C., Dunham, M. M., et al. 2018, *ApJ*, 853, 5
- Portegies Zwart, S. & McMillan, S. 2018, *Astrophysical Recipes : the art of AMUSE*, 2514-3433 (AAS IOP Astronomy publishing)
- Portegies Zwart, S., McMillan, S. L. W., van Elteren, E., Pelupessy, I., & de Vries, N. 2013, *Computer Physics Communications*, 184, 456

-
- Portegies Zwart, S. F., Makino, J., McMillan, S. L. W., & Hut, P. 1999, *A&A*, 348, 117
- Renaud, F., Gieles, M., & Boily, C. M. 2011, *MNRAS*, 418, 759
- Rice, T. S., Goodman, A. A., Bergin, E. A., Beaumont, C., & Dame, T. M. 2016, *ApJ*, 822, 52
- Shu, F. H. 1977, *ApJ*, 214, 488
- Shu, F. H., Adams, F. C., & Lizano, S. 1987, *ARA&A*, 25, 23
- Sills, A., Rieder, S., Scora, J., McCloskey, J., & Jaffa, S. 2018, *MNRAS*, 477, 1903
- Teyssier, R. 2002, *A&A*, 385, 337
- Tobin, J. J., Looney, L. W., Li, Z.-Y., et al. 2016, *ApJ*, 818, 73
- van de Voort, F., Schaye, J., Booth, C. M., Haas, M. R., & Dalla Vecchia, C. 2011, *MNRAS*, 414, 2458
- Vaytet, N., Audit, E., Chabrier, G., Commerçon, B., & Masson, J. 2012, *A&A*, 543, A60
- Vaytet, N., Chabrier, G., Audit, E., et al. 2013, *A&A*, 557, A90
- Vázquez-Semadeni, E., Palau, A., Ballesteros-Paredes, J., Gómez, G. C., & Zamora-Avilés, M. 2019, *MNRAS*, 490, 3061
- Wall, J. E., Mac Low, M.-M., McMillan, S. L. W., et al. 2020, *arXiv e-prints*, arXiv :2003.09011
- Wall, J. E., McMillan, S. L. W., Mac Low, M.-M., Klessen, R. S., & Portegies Zwart, S. 2019, *ApJ*, 887, 62
- Zari, E., Lombardi, M., Alves, J., Lada, C. J., & Bouy, H. 2016, *A&A*, 587, A106

MODÉLISER UN AMAS STELLAIRE AVEC AMUSE

Ce chapitre détaille la prise en main de la plateforme AMUSE à travers la simulation à N -corps d'un amas d'étoiles. AMUSE fût créé à la suite de la conférence MODEST-6, dont l'objectif était d'envisager les futures études traitant des amas d'étoiles (Davies et al., 2006). Les codes gravitationnels sont ainsi les plus nombreux à être intégrés à la plateforme. Ils offrent ainsi une entrée en matière privilégiée. De plus, le problème à N -corps est un des premiers à avoir été traités numériquement, nous détaillerons ainsi différents tests réalisés avec les codes d'AMUSE dans le but à la fois d'introduire le lecteur au mode de fonctionnement du logiciel mais aussi d'illustrer le comportement de ces codes face à diverses situations.

2.1 Introduction à AMUSE

L'idée pionnière d'AMUSE est avant tout de créer une plateforme unifiée, permettant de modéliser un problème astrophysique indépendamment du code utilisé pour le résoudre. Ainsi, AMUSE n'est pas un code de simulation mais bien un logiciel permettant d'utiliser de manière identique les codes qui y sont intégrés, via une interface ergonomique. Ces codes ne sont considérés que comme des modules interchangeable, appelés par le script principal guidant la simulation. La plupart de ces codes n'ont pas été développés par l'équipe d'AMUSE mais proviennent de l'ensemble de la communauté. On parle alors de codes communautaires. Ces derniers fonctionnent donc indépendamment de la plateforme, utilisant leur langage d'origine (C, C++, FORTRAN, etc.). En revanche AMUSE se charge de l'ensemble des communications entre les différents processus.¹ Ces communications sont invisibles pour l'utilisateur qui n'a donc pas à se soucier de ces détails techniques. Côté utilisateur, AMUSE apporte une surcouche logicielle à l'environnement PYTHON² (Van Rossum & Drake, 2009) et y intègre des outils dédiés pour mettre en place de manière ergonomique une simulation astrophysique. La Fig. 2.1 schématise le rôle d'interface que joue la plateforme.

En pratique, le script principal tout comme AMUSE, est écrit en PYTHON. Nous commencerons ce chapitre par initier le lecteur à mettre en place un tel script, de plus amples explications se trouvent en annexes A.1. Illustrons déjà deux des principales fonctionnalités implémentés au niveau de la structure des données. La première permet d'assigner une unité physique aux données (scalaires ou vectorielles), via l'opérateur « | » et ainsi de réaliser des calculs. À titre d'exemple :

```
# Les modules d'AMUSE s'importent comme des bibliothèques Python
In [1]: from amuse.units import units, constants
```

1. Ceci se fait via le protocole MPI (pour Message Passing Interface Gropp et al., 1996) ou SmartSOCKETS (Maassen & Bal, 2007), qui sont des standards implémentés sur la plupart des machines.

2. <https://www.python.org/>

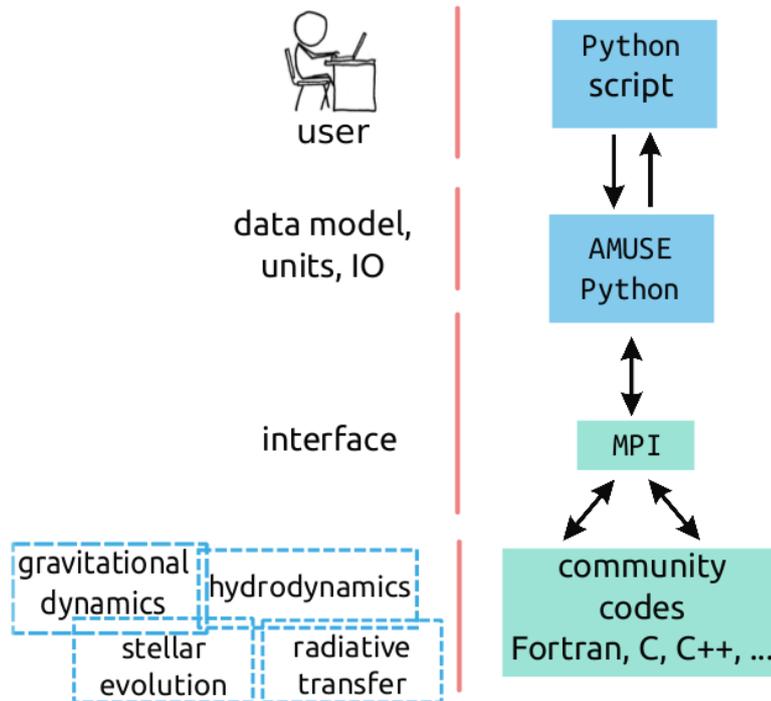


FIGURE 2.1 – Principe général de fonctionnement de la plateforme AMUSE.

```
In [2]: a_earth = constants.G * (1|units.MSun) / (1|units.AU**2)
```

`constants.G` est la constante gravitationnelle, `a_earth` donne l'accélération subie par la terre dans son orbite autour du soleil, c.-à-d. :

```
In [3]: print(a_earth)
```

```
Out [3]: quantity<0.0059316 * m * s**-2>
```

Deuxièmement, une simulation en astrophysique consiste à discrétiser le problème en un nombre fini d'éléments. Ces derniers peuvent prendre la forme de particules (dans une approche Lagrangienne) ou de cellules sur une grille (dans une approche Eulérienne). AMUSE fournit différentes structures (respectivement `Particles` ou `Grid`) dans le but de rassembler les paramètres physiques associés à ces données. En poursuivant l'exemple du système terre-soleil, on peut définir :

```
In [4]: from amuse.datamodel import Particles
```

```
In [5]: sun_earth = Particles(2)
```

```
In [6]: sun_earth[0].mass = 1|units.MSun
```

```
In [7]: sun_earth[1].mass = 1|units.MEarth
```

```
In [8]: sun_earth[0].position = [0, 0, 0]|units.AU
```

```
In [9]: sun_earth[1].position = [1, 0, 0]|units.AU
```

`sun_earth` représente la structure de donnée où sont stockés les attributs physiques des particules associées à la Terre et au Soleil. L'avantage est de pouvoir manipuler extrêmement facilement les données qui y sont associées par l'intermédiaire de méthodes dédiées, par exemple :

```
In [10]: sun_earth.center_of_mass()
Out [10]: quantity<[4.4920e+05, 0.0000, 0.0000][m]>
```

donne la position du centre de masse du système. Il est difficile de rendre compte de l'ensemble des possibilités que permettent ces structures. Ces dernières combinent les avantages d'objets PYTHON et leur manipulation se fait aussi simplement que des vecteurs de type NUMPY³ (Harris et al., 2020). Le but ici, n'est pas d'être exhaustif car le nombre de possibilités sont importantes. Plus d'exemples et de précisions peuvent être trouvés dans l'ouvrage *Astrophysical Recipes* (Portegies Zwart & McMillan, 2018). Ce livre ne constitue pas vraiment une documentation complète de la plateforme mais offre une vision plus globale de son fonctionnement et de ses possibilités. Ainsi la prise en main du logiciel par l'exemple constitue toujours la norme. Nombre d'entre eux peuvent être trouvés dans l'un des répertoires du logiciel. Notons déjà que cela constitue la force et la faiblesse du logiciel. La mise en place d'une simulation à partir d'un exemple est très aisée mais le manque de documentation ne permet pas de facilement se l'approprier et de la complexifier. Notons néanmoins qu'une documentation officielle et de plus en plus complète semble émerger et peut être trouvée ici.⁴ Il faut aussi garder en tête que le projet est collaboratif et est toujours en développement actif; les mises à jour sont donc fréquentes sur le dépôt github.⁵

2.2 Le problème à N -corps

2.2.1 Schémas d'intégration

Les étoiles sont des objets massifs, dont l'évolution est déterminée par l'attraction gravitationnelle qu'elles exercent entre elles. Dans le cas d'amas d'étoiles, la deuxième loi de Newton donne une solution pour calculer l'accélération \mathbf{a}_i subie par l'étoile i sachant les positions \mathbf{r}_j et les masses m_j des N étoiles environnantes :

$$\mathbf{a}_i = \sum_{j \neq i}^N \frac{Gm_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} (\mathbf{r}_i - \mathbf{r}_j), \quad (2.1)$$

où G est la constante de gravitation universelle et \mathbf{r}_i la position de l'étoile i .

Le problème à N -corps consiste à intégrer les équations du mouvement dans le but de prédire dans l'espace des phases les positions futures des étoiles composant le système :

$$\begin{cases} \mathbf{a}_i = \frac{d\mathbf{v}_i}{dt} \\ \mathbf{v}_i = \frac{d\mathbf{r}_i}{dt} \end{cases} \quad (2.2)$$

Ce système de N équations différentielles du second ordre et non linéaires ne possède pas de solution analytique pour $N > 3$. En revanche, l'approche numérique permet d'approximer la solution. Connaissant une configuration initiale, la méthode la plus largement utilisée consiste en une décomposition en séries de

3. <https://numpy.org>

4. <https://amuse.readthedocs.io/en/latest/>

5. <https://github.com/amusecode/amuse>

Taylor pour mettre à jour d'abord les vitesses (appelé *kick*) :

$$\mathbf{v}_{t+\Delta t} = \mathbf{v}_t + \Delta t \frac{d\mathbf{v}_t}{dt} + \mathcal{O}(\Delta t^2) = \mathbf{v}_t + \mathbf{a}(\mathbf{x}_t)\Delta t + \mathcal{O}(\Delta t^2) \quad (2.3)$$

puis les positions (*drift*) :

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \Delta t \frac{d\mathbf{x}_t}{dt} + \mathcal{O}(\Delta t^2) = \mathbf{x}_t + \mathbf{v}_{t+\Delta t}\Delta t + \mathcal{O}(\Delta t^2) \quad (2.4)$$

Le schéma numérique présenté ci-dessus s'arrête au premier ordre et l'erreur algorithmique associée est de $\mathcal{O}(\Delta t^2)$. Ainsi, il apparaît que le choix du pas de temps est déterminant dans le but d'approcher avec précision la solution. En revanche, diminuer le pas de temps nécessite de répéter l'intégration du système un plus grand nombre de fois pour évoluer le système jusqu'au même état donné. Ainsi, plus de précision demande soit un temps d'intégration plus long ou une machine plus puissante. La recherche d'un compromis est d'autant plus présente lorsque le nombre de particules du système devient important car chaque étape doit être répétée pour chacune d'entre elles. Le problème nécessite en effet d'évaluer l'accélération de N particules, qui elle-même requiert la somme de N éléments. Le nombre de calculs varie ainsi en N^2 , devenant vite prohibitif notamment dans un contexte astrophysique.

L'histoire des simulations du problème à N -corps a commencé il y a plus de 50 ans, lorsque les ordinateurs n'était capable que de quelques milliers d'opérations par seconde. Ainsi de nombreuses solutions ont été développées dans le but de permettre un choix de pas de temps plus grand sans augmenter drastiquement le temps, ni l'imprécision du calcul. La méthode du *Leap-frog* (Verlet, 1967) a été précurseur dans le but d'améliorer la stabilité de l'intégration. Elle consiste simplement à évaluer les vitesses au milieu du pas de temps en utilisant les informations déjà calculées au pas de temps précédent :

$$\mathbf{v}_{t+\frac{\Delta t}{2}} = \mathbf{v}_t + \frac{\Delta t}{2} \mathbf{a}(\mathbf{x}_t) \quad (2.5)$$

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \Delta t \mathbf{v}_{t+\frac{\Delta t}{2}} \quad (2.6)$$

$$\mathbf{v}_{t+\Delta t} = \mathbf{v}_{t+\frac{\Delta t}{2}} + \frac{\Delta t}{2} \mathbf{a}(\mathbf{x}_{t+\Delta t}) \quad (2.7)$$

Ce schéma d'intégration du second ordre est dit symplectique. Par construction, cette propriété limite la dérive en énergie de l'intégration (Hairer et al., 2003). Il est encore utilisé, notamment pour des systèmes non-collisionnels où le potentiel généré par les particules est relativement lisse. Pour des systèmes collisionnels, comme les amas d'étoiles, les variations du potentiel gravitationnel sont plus importantes, nécessitant une meilleure évaluation de l'accélération subies par les étoiles. Ainsi ont vu le jour des intégrateurs d'ordre supérieur se développer. Ces derniers utilisent des dérivés d'ordre supérieur raffinant l'évaluation de l'espace des phases à chaque pas de temps. Les schémas du 4^e ordre utilisant les polynômes de Hermite sont aujourd'hui les plus populaires, notamment grâce à son implémentation la plus connue sous NBODY6 (Aarseth, 1999). Si les premiers calculs étaient autrefois limités à quelques dizaines de particules, notons qu'aujourd'hui cette approche peut être adoptée pour traiter des amas comportant jusqu'à environ 1 millions de particules (Wang et al., 2016).

2.2.2 Unités Hénon et équilibre du viriel

La plupart des codes d'intégration numérique n'utilisent pas les unités du système international (SI). Dans le but d'optimiser le calcul, les données du problèmes sont adimensionnées et normalisées. Pour les simulations N -corps, la convention des unités Hénon (Heggie & Mathieu, 1986) est depuis longtemps utilisée dans le but de faciliter la comparaison entre différents codes ou études :

$$\begin{cases} G = 1 \\ M = 1 \\ E = -\frac{1}{4}, \end{cases} \quad (2.8)$$

où G est la constante de gravitation, M la masse totale du système et E l'énergie totale du système. Ce choix simplifie les grandeurs manipulées, en particulier lorsque l'on traite un système en isolation et à l'équilibre. En effet, un système irrotationnel ne subissant que la gravité des particules qui le compose (dit auto-gravitant) est à l'équilibre du viriel lorsque :

$$2E_k + E_p = 0, \quad (2.9)$$

où E_k est l'énergie cinétique et E_p l'énergie potentielle du système. Avec la convention utilisée plus haut, on obtient :

$$\begin{cases} E_k = \frac{1}{4} \\ E_p = -\frac{1}{2} \end{cases} \quad (2.10)$$

En définissant une formulation globale de l'énergie potentielle du système de la forme :

$$E_p = -\frac{GM^2}{2R_v}, \quad (2.11)$$

on obtient $R_v = -\frac{GM^2}{4E} = 1$, une distance caractéristique appelée rayon du viriel. De la même manière, l'énergie cinétique globale du système est définie en fonction d'une vitesse moyenne au carré des particules, permettant de définir l'unité de temps du système Hénon :

$$U_t = \frac{GM^{\frac{5}{2}}}{(-4E)^{\frac{3}{2}}}. \quad (2.12)$$

L'avantage de ce système d'unités réside dans le fait qu'il n'est pas complètement contraint, permettant ainsi de mettre à l'échelle souhaitée n'importe quel problème. Il faut donc spécifier la masse totale M et le rayon du viriel R_v (ou l'unité de temps associée) en unité SI pour pouvoir réaliser la conversion. `AMUSE` fournit une classe d'objet dédiée et permettant de faciliter ces conversions. Ainsi un système de masse $M = 1000 M_\odot$ et de rayon $R_v = 1$ pc aura un temps caractéristique de 0.86 M.a..

```
In [1]: from amuse.units import units, nbody_system
In [2]: convertir = nbody_system.nbody_to_si(1000 | units.MSun, 1|units.pc)
In [3]: print(convertir.to_si(1|nbody_system.time))
```

```
Out [3]: quantity<0.8607[Myr]>
```

2.3 Tests des codes gravitationnels

Les différents codes intégrés sous AMUSE sont issus de sources diverses. Leur bon fonctionnement est fonction de la qualité de leur intégration à la plateforme. Il n'est ainsi pas rare qu'un code puisse « crasher » alors qu'il semble correctement installé. De plus, il est ainsi souvent difficile d'avoir accès à leur paramétrage précis car la documentation d'AMUSE est succincte sur ce point et la version implémentée ne correspond pas forcément à la description des articles de références. Il est alors crucial d'effectuer des tests opérationnels permettant d'appréhender leur fonctionnement et leurs performances. On ne se concentrera pas sur les détails techniques mais plutôt sur les forces et faiblesses de certains d'entre eux dans différentes situations.

2.3.1 Mise en place de conditions initiales

Pour débiter une simulation il faut d'abord faire le choix de conditions initiales. Dans le cas d'amas d'étoiles, le modèle de Plummer (Plummer, 1911) fait figure de référence. Ce modèle représente en effet une solution analytique stable à l'équation de Poisson. Ce modèle représente un amas ayant une densité uniforme en son centre et diminuant en $1/R$. Son potentiel est défini par :

$$\Phi = -\frac{GM}{\sqrt{r^2 + b^2}}, \quad (2.13)$$

où le paramètre b définit le rayon de cœur de l'amas. Bien que de masse finie, ce modèle s'étend à l'infini. Ainsi ce modèle ne décrit pas fidèlement les amas observés. En revanche, la forme analytique de son potentiel rend sa mise en place aisée et c'est un choix depuis longtemps adopté dans les simulations aux N -corps. Pour générer ce type de modèle avec AMUSE :

```
In [1]: from amuse.ic.plummer import new_plummer_model
In [2]: cluster = new_plummer_model(1000)
```

L'objet `cluster` représente un jeu de particules distribuées selon un tel modèle. On peut afficher les données de ce set et correspondant donc à l'espace des phases du système :

```
In [3]: print(cluster)
```

```
Out [3]:
```

key	mass	radius	x	y	z	vx	vy	vz
-	mass	length	length	length	length	length/time	length/time	length/time
13593262122530608750	0.001	0.	0.536	-0.024	-0.003	1.155	-0.177	0.125
453850965235164268	0.001	0.	0.859	0.364	0.962	0.656	0.144	0.491
9049866169888506120	0.001	0.	0.390	0.125	0.579	0.052	-0.315	-0.204
420964685199832469	0.001	0.	0.824	-0.56	0.227	-0.564	0.363	-0.120
...

Pour une visualisation plus imagée, on peut porter en graphique la position des étoiles en utilisant p. ex. `matplotlib`⁶ (Hunter, 2007, voir Fig. 2.2) :

6. <https://matplotlib.org/>

```

In [4]: import matplotlib.pyplot as plt
In [5]: fig, ax = plt.subplots()
In [6]: ax.scatter(cluster.x.number, cluster.y.number)
In [7]: fig.show()

```

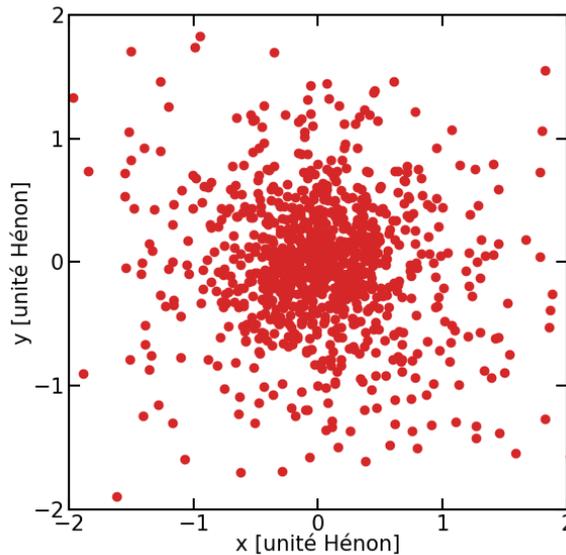


FIGURE 2.2 – Visualisation d'un modèle de Plummer.

2.3.2 Intégration d'un modèle de Plummer à l'équilibre

Le premier test que j'ai réalisé consiste à évaluer la précision d'une intégration d'un modèle de Plummer de 1000 particules à l'équilibre pendant 1 temps Hénon (l'annexe A.1.5 détaille le script qui a été utilisé pour réaliser ce test). L'énergie totale d'un tel système devant être conservée, on utilisera cette mesure pour évaluer la précision de différents codes de la plateforme. Les codes testés sont *Hermite* (Hut et al., 1995), *Phi-GRAPe* (Harfst et al., 2007) et *ph4*, tous trois des intégrateurs se basant sur un schéma du 4^e ordre utilisant les polynômes de Hermite (Makino, 1991); *Huayno*, un intégrateur symplectique du 2^e ordre (Jänes et al., 2014; Pelupessy et al., 2012) et *Sakura*, une méthode se basant sur une décomposition en problèmes à deux corps dont chaque solution est exacte. La composition de l'ensemble produit une approximation au 2^e ordre (Gonçalves Ferrari et al., 2014). Comme on l'a vu au §2.2, le paramètre principal qui va déterminer la précision du calcul est celui définissant le pas de temps. La Fig. 2.3 montre la dérive en énergie en fonction de ce paramètre. Pour un même paramètre de pas de temps, la précision du calcul est différente même pour des codes utilisant des algorithmes similaires. Ce paramètre, réglé au niveau du script principal d'AMUSE, ne désigne pas exactement la même chose selon le code utilisé. Nous nous intéresserons donc à la forme de chaque courbe, plus ou moins indépendamment de leur position relative. Sur le panneau de gauche, on observe comme attendu, qu'une diminution du pas de temps conduit à améliorer la précision du calcul. En revanche, ceci se fait au dépend d'un nombre croissant d'opérations, ralentissant ainsi l'intégration (panneau de droite).

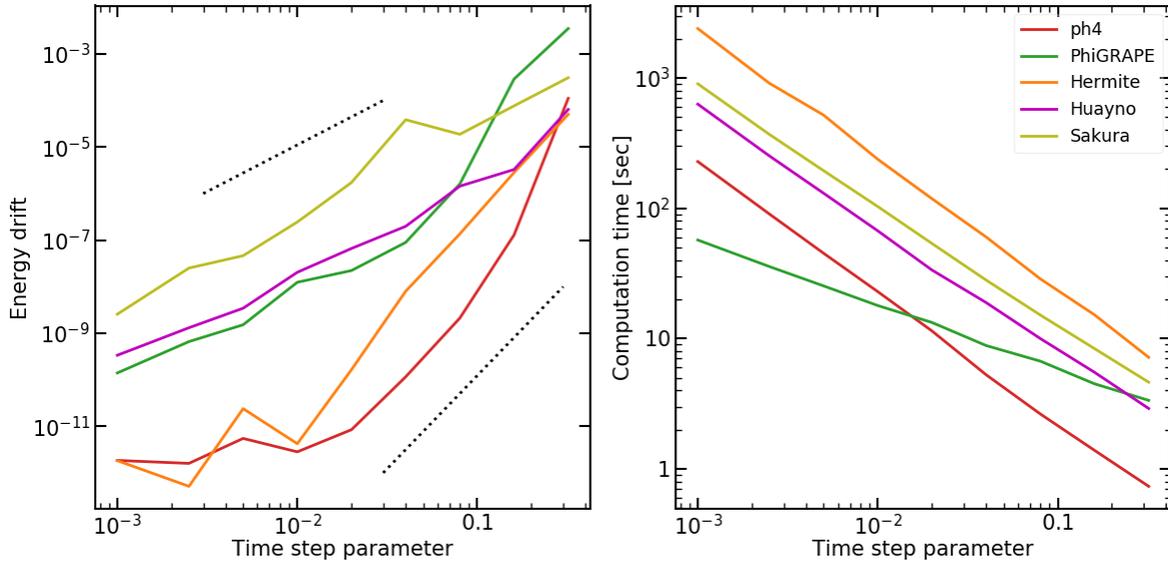


FIGURE 2.3 – Panneau de gauche : dérive en énergie $\frac{|E_{\text{début}} - E_{\text{fin}}|}{E_{\text{début}}}$ lors de l’intégration d’un système de Plummer de 1000 particules sur un temps de 1 unité Hénon en fonction du paramètre de pas de temps. Les droites en pointillées représente une amélioration de la précision de calcul proportionnelle à Δt^2 et Δt^4 pour celle situé en haut et en bas respectivement. Panneau de droite : augmentation du temps de calcul (en secondes) due à la diminution du pas de temps de l’intégration.

L’amélioration de la précision de l’intégration ne se fait pas au même rythme selon le code choisi (voir §2.2. La droite en pointillée au bas du panneau de gauche montre une amélioration de la précision proportionnelle à Δt^4 , correspondant globalement aux intégrateurs du 4^oordre. À contrario, la droite en pointillée en haut de la figure indique une amélioration en Δt^2 correspondant aux intégrateurs du 2^oordre (c.-à-d. Huayno, Sakura). Alors que les intégrateurs du 4^oordre atteignent rapidement une précision maximale dictée par la machine (à l’exception surprenante de PhiGRAPE, les autres nécessitent des pas de temps beaucoup plus petits pour atteindre cette même précision, au prix d’une explosion du temps de calcul.

La précision maximale d’une machine est dictée par la représentation des nombre réels en un nombre fini de bits. Sur une machine 64 bits, PYTHON utilise par défaut le formatage des nombres réels en virgule flottante double précision. Cette représentation permet de distinguer deux nombres réels dont la différence est supérieur à $2^{-53} = 1.1 \times 10^{-16}$. On remarque néanmoins qu’une telle précision n’est pas atteinte dans l’exemple ci-dessus. Hermite et ph4 atteignent un maximum de précision d’environ 10^{-13} pour un paramètre de pas de temps approchant 10^{-2} . La principale différence entre ces deux codes se situe au niveau de la rapidité d’intégration, ph4 étant dix fois plus rapide qu’Hermite. Étonnamment, la précision de PhiGRAPE, utilisant le même schéma que les deux codes précédent, ne progresse plus en Δt^4 lorsque le paramètre de pas de temps est inférieur à 0.1. En revanche, le temps de calcul associé augmente environ deux fois moins vite. Ces petites différences sont intimement liées à la structure du code effectuant le calcul et dont AMUSE n’a pas forcément de prise. De plus, la surcouche qu’apporte le logiciel dans l’interfaçage des codes sources avec la plateforme peut rendre leur fonctionnement plutôt opaque. C’est un point problématique déjà soulevé, qui en l’absence

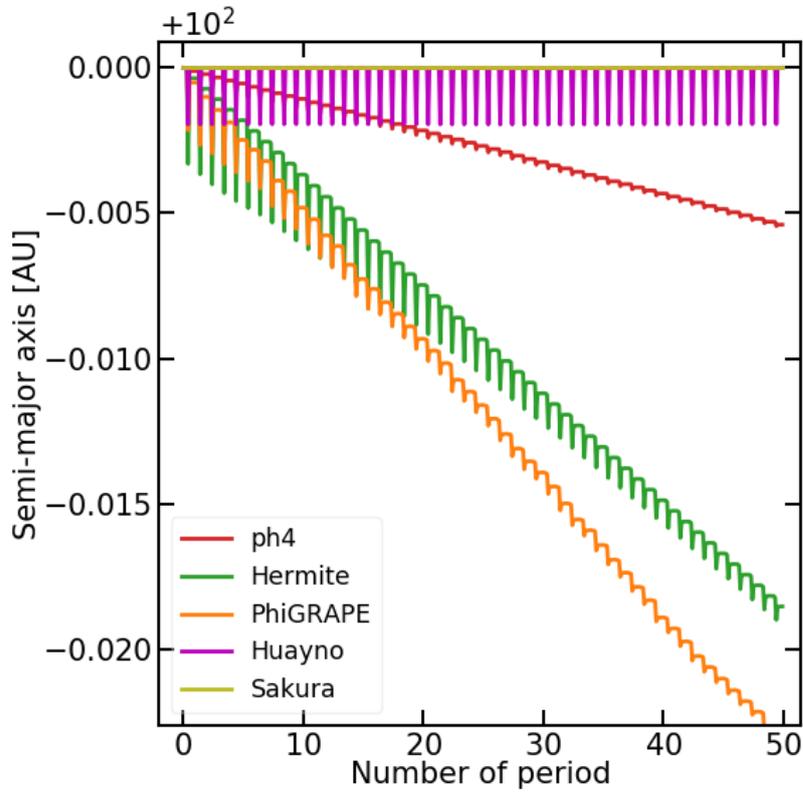


FIGURE 2.4 – Variation de la distance à l'apogée entre les composantes d'un système binaire ayant pour semi-grand axe majeur 100 AU et d'excentricité $e = 0.99$ sur une période d'intégration de 50 périodes de révolution, avec différents intégrateurs d'AMUSE.

de documentation technique ne permet pas un paramétrage au plus juste.

2.3.3 Intégration d'une binaire excentrique

Le comportement chaotique du problème à N -corps provient principalement de la présence d'une singularité dans le calcul de l'accélération que subit chaque particule. En effet, cette accélération diverge lorsque deux particules se rapprochent fortement. Ceci constitue une difficulté pour toute intégration numérique. Une illustration simple peut être réalisée via l'intégration d'un système binaire très excentrique. En effet, la variation d'accélération Δa au cours d'une révolution d'un tel système varie entre le passage du périhélie à l'apogée d'un facteur $(\frac{1+e}{1-e})^2$ où e est l'excentricité orbitale. Pour une orbite elliptique où $e = 0.99$, $\Delta a \sim 10^4$. La Fig. 2.4 montre la variation du demi-grand axe majeur d'une binaire d'excentricité $e = 0.99$ au cours de 50 périodes de révolution, en utilisant les mêmes codes que précédemment. Un pas de temps petit et correspondant à un temps de calcul similaire est utilisé.

Dans ces conditions, les différents codes intègrent avec difficulté chaque passage au périhélie, à l'exception de Sakura qui par construction retrouve la solution exacte du problème à deux corps. En ce qui concerne les intégrateurs de Hermite (c.-à-d. ph4, Hermite et PhiGRAPE), on observe une dérive séculaire du demi-

grand axe majeur de la binaire, conduisant à un resserrement de son orbite. Cette erreur augmente linéairement et atteint 1% après 100 révolution pour le plus précis d'entre eux (c.-à-d. `ph4`). Cela correspond pour une binaire de type solaire à une évolution d'environ 0.1 M.a.. Cela n'est pas négligeable si l'on veut rendre compte des propriétés des systèmes binaires même au sein d'amas jeunes de quelques millions d'années. Ce problème sera d'ailleurs traité dans le chapitre 6 où l'on verra une méthode pour pallier cette difficulté. À l'opposé, on remarque que `Huayno` bien que d'ordre inférieur ne dérive pas et reste globalement proche de la solution exacte. C'est une propriété intéressante de l'algorithme symplectique utilisé (de type *leapfrog*) qui pour des systèmes peu chaotiques conservent bien les intégrales du mouvement.

2.3.4 Interactions fortes au sein d'un amas

Les interactions fortes étoile-étoile particulièrement difficile à intégrer dans une simulation aux N -corps ne se produisent pas qu'au sein de systèmes binaires. À l'intérieur d'amas denses, ces interactions à courte distance peuvent être courantes entre deux étoiles non liées (McMillan, 2015). Dans le but de représenter les processus dynamiques à l'œuvre, les codes doivent correctement intégrer les échanges d'énergie issus de ces interactions. Ainsi un dernier test que nous avons réalisé consiste à produire une rencontre frontale entre deux étoiles au sein d'un amas. Les étoiles n'ont pas de taille physique dans cet exemple, on considère seulement la déflexion des trajectoires due à l'interaction gravitationnelle. Cette situation extrême permettra d'estimer l'erreur maximale réalisée sur un pas de temps d'évolution du système. La configuration initiale correspond toujours à une sphère de Plummer à l'équilibre. Dans un tel système cet équilibre n'est défini que par l'énergie des particules qui le composent et non par la direction des vecteurs vitesses des étoiles qui est donc choisie aléatoirement. Nous avons mis en place une méthode pour diriger les vecteurs vitesse des deux étoiles les plus proches du centre de l'amas l'un vers l'autre dans le but de les voir se rencontrer. Le script en annexe A.2 détaille la fonction utilisée en utilisant les fonctionnalités d'AMUSE. On intègre ce système avec les deux intégrateurs les plus précis du §2.3.2, c.-à-d. `ph4` et `Hermite`. On teste aussi le mode GPU permis par le code `ph4` qui promet une accélération de l'intégration avec l'utilisation d'une carte GPU. La Fig. 2.5 montre l'évolution de l'énergie totale du système pendant l'interaction pour deux paramètres de pas de temps égaux à 0.05 et 0.02. On observe qu'au moment de la rencontre à $t \approx 0.017$, l'énergie du système marque un saut en énergie égale à plus de 10 fois ce qui est observé dans les conditions idéales de la Fig. 2.3. Cela montre tout d'abord que ces interactions jouent un rôle prépondérant dans la précision de l'intégration. De plus, on remarque que les courbes vertes correspondant au mode GPU de `ph4` n'atteignent pas la précision de celles en rouge du mode CPU. Ainsi bien que ce code conduise à une légère accélération de l'intégration en mode GPU⁷, ce dernier ne permet pas d'atteindre une précision supérieure à 10^9 . Pire, en forçant le paramètre de pas de temps en dessous de 0.02, la durée du calcul s'envole sans influence sur la précision. Si la raison de ce comportement est restée pendant longtemps ambiguë, il semble aujourd'hui que cela dépasse la seule responsabilité du logiciel. En effet, nous avons remarqué que les différentes cartes GPU que nous avons utilisées n'adoptaient qu'un formatage des nombres en simple précision, ce qui limite la précision des calculs à $\approx 2^{-24} = 5.9 \times 10^{-8}$, ce qui correspond approximativement à ce qui est observé ici.

7. Elle n'est toutefois vraiment intéressante lorsque le nombre de particules dépasse 10^4 .

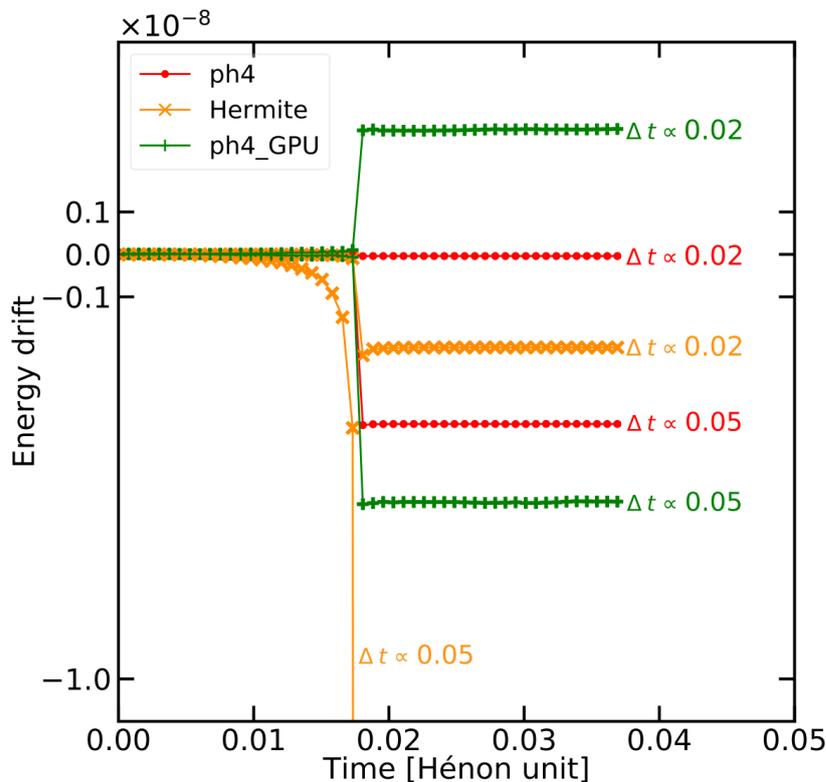


FIGURE 2.5 – Dérive en énergie $\frac{|E_0 - E_t|}{E_0}$ en fonction du temps d'un système de Plummer de 1000 étoiles, lors d'une interaction frontale entre deux étoiles (le paramètre d'impact tend vers 0) à $t \approx 0.017$ temps Hénon. Deux réalisations sont effectuées correspondant à deux pas de temps différents pour chaque intégrateur : `ph4` (lignes rouges), `Hermite` (lignes oranges) et `ph4_GPU` qui est l'implémentation de `ph4` sur GPU (lignes vertes). On remarque que ce dernier code n'atteint pas la même précision que l'implémentation sur CPU à partir d'un paramètre de pas de temps < 0.02 .

2.4 Conclusion

Nous avons dans ce chapitre, abordé le problème à N -corps via une approche numérique et grâce à la plateforme AMUSE. Cela nous a permis de prendre en main la plateforme mais aussi de tester les codes pertinents dans le but d'intégrer des amas d'étoiles jeunes. Cela a aussi permis de choisir le code `ph4` pour la suite du travail. Il représente en effet le meilleur compromis précision-temps de calcul des codes testés. Le paramétrage vis à vis du pas de temps d'intégration a été choisi de manière à ce que la dérive en énergie totale d'une simulation ne dépasse pas 0.1%. De manière générale cette dernière est dominée par les quelques (rares) interactions fortes au sein du système. Ainsi l'erreur associée ne concerne donc qu'un nombre limité d'orbite. En raison de ce critère, nous n'avons en général pas utilisé le mode GPU de `ph4` en raison de sa limite en précision explicitée à la section 2.3.4

Nous nous intéressons dans le chapitre suivant à la méthode pour reproduire des distributions d'amas fragmentés ainsi qu'aux outils d'identification de surdensités stellaires. Ils constitueront notre base à l'analyse des processus dynamiques à l'œuvre au sein des amas jeunes.

BIBLIOGRAPHIE

- Aarseth, S. J. 1999, *PASP*, **111**, 1333
- Davies, M. B., Amaro-Seoane, P., Bassa, C., et al. 2006, , **12**, 201
- Gonçalves Ferrari, G., Boekholt, T., & Portegies Zwart, S. F. 2014, *MNRAS*, **440**, 719
- Gropp, W., Lusk, E., Doss, N., & Skjellum, A. 1996, *Parallel Computing*, **22**, 789
- Hairer, E., Lubich, C., & Wanner, G. 2003, *Acta Numerica*, **12**, 399
- Harfst, S., Gualandris, A., Merritt, D., et al. 2007, , **12**, 357
- Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, *Nature*, **585**, 357–362
- Heggie, D. C. & Mathieu, R. D. 1986, Standardised Units and Time Scales, ed. P. Hut & S. L. W. McMillan, Vol. 267 (Springer Berlin Heidelberg), 233
- Hunter, J. D. 2007, *Computing in Science and Engineering*, **9**, 90
- Hut, P., Makino, J., & McMillan, S. 1995, *ApJ*, **443**, L93
- Jänes, J., Pelupessy, I., & Portegies Zwart, S. 2014, *A&A*, **570**, A20
- Maassen, J. & Bal, H. E. 2007, in *Proceedings of the 16th IEEE International Symposium on High-Performance Distributed Computing (HPDC), Monterey, CA, USA*, 1–10
- Makino, J. 1991, *ApJ*, **369**, 200
- McMillan, S. L. W. 2015, Dynamical Processes in Globular Clusters, ed. H. M. J. Boffin, G. Carraro, & G. Beccari, Vol. 413, 225
- Pelupessy, F. I., Jänes, J., & Portegies Zwart, S. 2012, , **17**, 711
- Plummer, H. C. 1911, *MNRAS*, **71**, 460
- Portegies Zwart, S. & McMillan, S. 2018, *Astrophysical Recipes : the art of AMUSE*, 2514-3433 (AAS IOP Astronomy publishing)
- Van Rossum, G. & Drake, F. L. 2009, *Python 3 Reference Manual* (Scotts Valley, CA : CreateSpace)
- Verlet, L. 1967, *Physical Review*, **159**, 98
- Wang, L., Spurzem, R., Aarseth, S., et al. 2016, *MNRAS*, **458**, 1450

IDENTIFICATION DE STRUCTURES AU SEIN D'AMAS FRAGMENTÉS

Dans ce chapitre, nous nous intéresserons à la méthode de fragmentation à N -corps développée par [Dorval et al. \(2016\)](#) dans le but de construire de manière auto-cohérente des distributions fragmentées d'étoiles représentatives de régions de formation stellaire. Ces modèles seront reproduits avec l'aide de la plateforme AMUSE. La diversité des environnements observés au sein des régions jeunes a conduit à développer de nombreux outils statistiques dans le but de caractériser leur structure. Nous expliciterons deux méthodes que nous avons utilisées sur ces modèles : l'algorithme HOP et le Minimum Spanning Tree (MST). Ces derniers ont été choisis car ils permettent non seulement d'identifier les agrégats les plus denses d'une distribution quelconque de points, mais aussi de caractériser sa structure globale.

3.1 Modèle de fragmentation aux N -corps

La fragmentation d'un nuage moléculaire consiste en l'imbrication de multiples processus physiques non-linéaires. La modélisation de l'ensemble du processus représente un défi pour les simulations en raison des échelles impliqués. En effet, la densité du gaz au sein d'un GMC de rayon de 10 pc et de masse $1000 M_{\odot}$ traverse 22 ordres de grandeur pour atteindre la densité d'une étoile de type solaire. Diverses approximations sont donc adoptées pour réaliser ces calculs, notamment aux plus petites échelles. La recette la plus utilisée dans le but de pallier la limite en résolution, consiste à introduire des particules de substitution, dites *sink particles* ([Bate et al., 1995](#)), aux endroits où le gaz atteint une densité critique considérée suffisante pour la formation d'une proto-étoile. La densité choisie se situe généralement autour de 10^{-7} g/cm^3 ce qui correspond à la formation du second cœur de Larson (dans les dernières simulations de [Bate \(2019\)](#), elle est même de 10^{-5} g/cm^3). Ces *sink particles*, n'étant plus régies par les lois de l'hydrodynamique, approximent donc les étoiles en formation en train d'accréter la matière qui les entourent. Cette approximation a permis l'important développement des simulations hydrodynamiques à des régions de quelques fractions de parsecs à une résolution de l'ordre de l'U.A. (voir p. ex. [Federrath et al., 2010](#), pour une étude sur le bon usage de la méthode). Néanmoins, ces simulations produisent seulement quelques centaines de jeunes étoiles au sein de petits volumes. Leur coût très important en ressource de calcul ne permet donc pas d'explorer un grand nombre de configurations, ni leur évolution à plus long terme ou en lien avec un environnement élargi.

À contrario, l'approche N -corps permet d'explorer de nombreuses distributions différentes comportant quelques milliers à centaines de milliers d'étoiles pouvant représenter des régions entières. Cette approche fait en général l'impasse sur les processus de formation des étoiles. Le choix des conditions initiales détermine alors

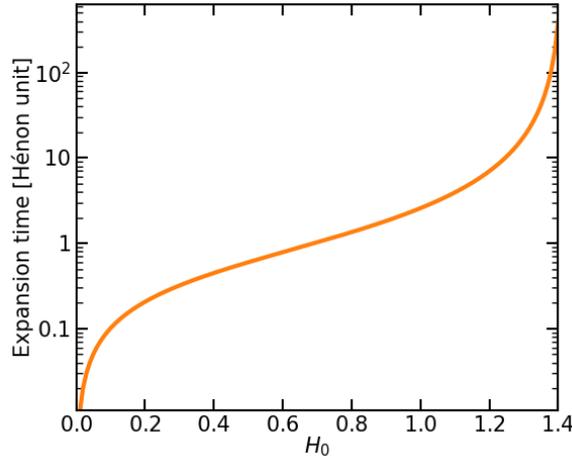


FIGURE 3.1 – Variation de la durée de la phase d'expansion d'un modèle GDF en fonction du paramètre H_0 .

grandement les résultats obtenus. La distribution spatiale des étoiles à leur naissance joue pourtant un rôle déterminant dans l'évolution dynamique de l'amas. Cette dernière est largement issue de la structure fragmenté du gaz. Dorval et al. (2016) ont proposé une méthode simple et efficace dans le but de reproduire ce processus de fragmentation seulement guidé par la gravité. Elle consiste à réaliser une détente adiabatique d'un système à N -corps auto-gravitant. En pratique, on initie une distribution sphérique uniforme de particules que l'on lance dans une expansion homothétique. Ainsi le champ des vitesses est dirigé à l'opposé du centre de la sphère et la norme du vecteur vitesse de chaque particule est proportionnelle à sa distance au centre :

$$\mathbf{v}_0 = H_0 \mathbf{r}_0 \tag{3.1}$$

où \mathbf{v}_0 et \mathbf{r}_0 correspondent respectivement aux vecteurs vitesses et positions initiaux des particules, H_0 est une constante positive. Au sein de ce système en expansion, on permet aux petites fluctuations poissonniennes de densité présentes dans la distribution initiale de se développer seulement par attraction gravitationnelle. Cela conduit à agréger les particules entre elles autour des puits de potentiel les plus significatifs. De cette manière, les particules du système s'agencent en un réseau de nœuds et de filaments caractéristiques de la fragmentation observée dans les régions de formation stellaire. Cette méthode n'est pas sans rappeler la croissance des structures aux échelles cosmologiques au début de l'univers, d'où la dénomination initiale de Dorval et al. (2016) en modèles de Hubble-Lemaître. Néanmoins, pour éviter toute confusion, nous renommerons cette méthode GDF pour *Gravity-Driven Fragmentation*. En effet une des différences majeures vient du fait que l'expansion des modèles GDF n'est pas alimentée et ralentit du fait de l'attraction gravitationnelle. En effet, c'est la valeur de H_0 qui détermine la force de l'expansion et ainsi l'état final du système. L'étude analytique de Dorval et al. (2016) (§ 2 ainsi que leur annexe B) permet de déterminer le temps d'expansion d'un tel système pour une valeur de H_0 donné. La Fig. 3.1 montre la variation de la durée de l'expansion selon le paramètre H_0 . On observe que pour une valeur $H_0 > \sqrt{2}$ la durée d'expansion du système diverge et le système s'étend donc à l'infini. Puisque l'on s'intéresse à des systèmes liés, on choisira des valeurs $H_0 < \sqrt{2}$.

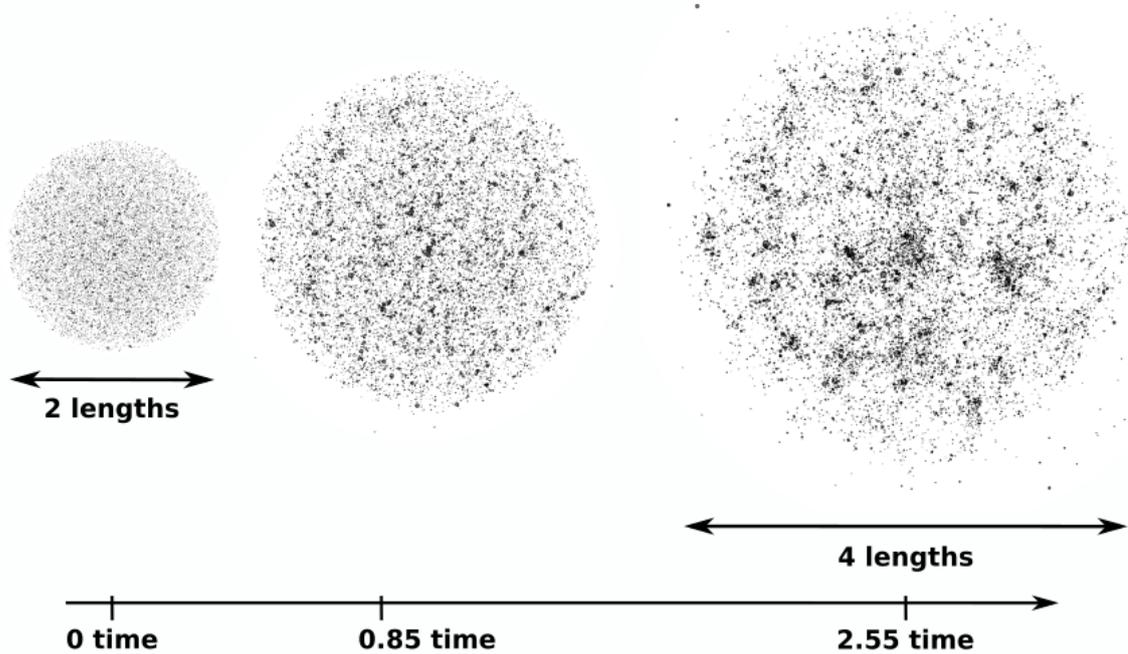


FIGURE 3.2 – Fragmentation progressive d'un modèle GDF en unité Hénon. À gauche est représenté l'état initial : les étoiles sont uniformément distribuées et le champ des vitesses est purement radial. Au milieu, le système s'étend et permet aux fluctuations de densités de se développer. À droite, le modèle est maintenant fragmenté et l'expansion s'arrête.

Une autre différence importante entre les modèles GDF et les simulations cosmologiques vient aussi du fait que l'on considère les particules du modèle comme des étoiles déjà formées. Elles suivent donc un spectre de masse. [Dorval et al. \(2016\)](#) ont étudié en détail la formation des petits agrégats ou clumps au sein des modèles GDF en fonction de l'IMF choisie. Ils ont en particulier montré que ces derniers se formaient autour des étoiles massives. De cette façon, la fonction de masse des étoiles au sein des agrégats définie comme une loi de puissance $dN/dm \propto m^{-\alpha}$ possède un index de $\alpha \approx 2.15$ moins élevé que la valeur de référence ~ 2.35 ([Salpeter, 1955](#)) en cohérence avec les observations au sein de la Voie Lactée ([Bastian et al., 2010](#)). De plus, la formation des clumps se faisant de manière progressive, ils ont montré que les étoiles massives se retrouvaient préférentiellement au centre des agrégats. Cette ségrégation en masse primordiale des clumps s'accordent ainsi avec les résultats de simulations hydrodynamiques ([Maschberger et al., 2010](#)).

La Fig. 3.2 montre les étapes de formations d'un de ces modèles comportant $N = 15\,000$ étoiles distribuées selon une IMF de [Maschberger \(2013\)](#) (reproduisant fidèlement la forme de l'IMF de [Kroupa 2001](#) et [Chabrier 2003](#)) et initialisé avec un paramètre $H_0 = 1$. Un tel modèle peut être reproduit grâce au scripts en annexe A.3.1 ou A.3.2. Sur une machine de bureau, l'intégration pour $N \sim 1000$ dure une quinzaine de minutes, et plusieurs heures pour $N \sim 10000$.

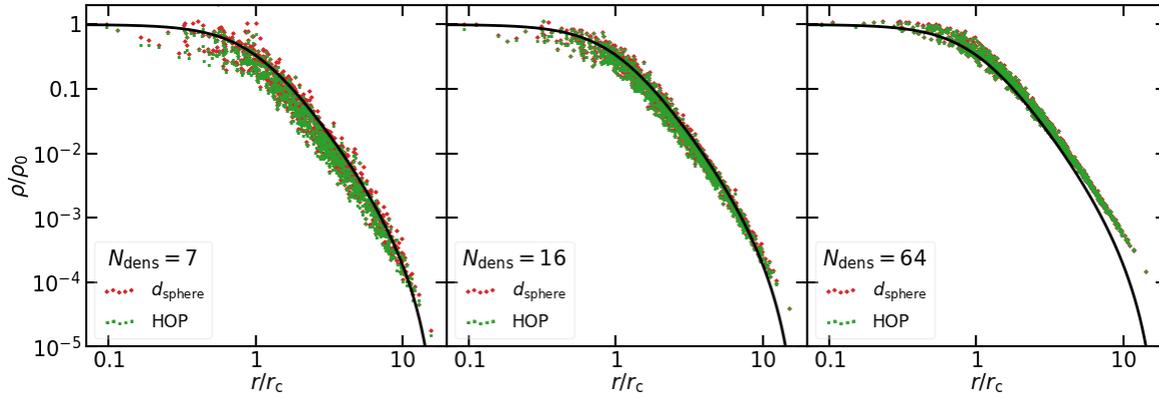


FIGURE 3.3 – Profil radial de densité d'un amas de 1000 étoiles distribuées selon un modèle de King de paramètre $W_0 = 6$. L'estimation de la densité locale de chaque particule par la méthode HOP et selon l'Eq. 3.2 est représentée respectivement en vert et en rouge.

3.2 Algorithme de détection de surdensités

Si les observations des dernières années indiquent que les étoiles se forment au sein d'une large variété d'environnements de densités différentes (Gouliermis, 2018), l'étude des régions les plus denses est toujours d'un intérêt particulier. Elles sont en effet les sites où la transformation du gaz en étoiles est la plus efficace et ainsi le lieu privilégié de la formation d'étoiles massives (Motte et al., 2018). Elles sont aussi souvent considérées comme les éléments précurseurs d'amas ouverts plus âgés car mieux capables de résister à l'expulsion du gaz (Krumholz, 2014). Les modèles GDF reproduisent ces surdensités en cohérence avec leur environnement. Ils offrent donc un banc d'essai incontournable dans le but de tester différents outils de détection.

3.2.1 L'algorithme HOP

La méthode HOP Eisenstein & Hut (1998) est originellement un algorithme de détection de halo de matière noire dans les simulations cosmologiques. Son application convient particulièrement aux amas d'étoiles car les configurations fragmentées qu'elles arborent sont assez similaires à celles de la toile cosmique structurées par la matière noire à grande échelle. De plus, cela permet à l'algorithme d'être très efficace sur de grands jeux de données.

Comme la plupart des algorithmes de détection, HOP n'impose pas de système de coordonnées mais suit la distribution des particules. Ainsi, son principe de fonctionnement se base sur l'estimation de la densité locale assignée à chaque particule. Cette estimation est réalisée à partir de la distribution des plus proches voisins de chaque particule. Les pics de densités sont identifiés et les particules environnantes y sont associées par l'ajustement de paramètres agissant comme des seuils. Le choix de ces paramètres permet d'ajuster sur des critères physiques la sélection des zones d'intérêts. Les paramètres par défaut dans l'étude originelle de Eisenstein & Hut (1998) sont calibrés pour de grandes simulations cosmologiques. Nous avons donc exploré l'espace des paramètres pour une application aux modèles GDF.

Estimateur de densité

L'estimateur de densité de HOP est un atout majeur de la méthode. À chaque particule est calculée une densité se basant un nombre N_{dens} de ses voisins. Un noyau de lissage est utilisé pour adapter l'estimation à la configuration locale. Le paramètre N_{dens} laissé au choix de l'utilisateur permet d'adapter l'intensité du lissage. Un premier test est réalisé sur un amas de 1000 particules distribuées selon un modèle de King (King, 1962). La Fig. 3-3 montre cette estimation de densité pour chacune des particules de l'amas (nuage de points verts). Les différents panneaux correspondent à différentes valeurs de N_{dens} . En comparaison, une deuxième estimation est réalisée sur la sphère englobant le même nombre de voisins, et égale à

$$d_N = \frac{3 \sum_i^N M_i}{4\pi r_N^3}, \quad (3.2)$$

où M_i sont les masses des N plus proches voisins et r_N , la distance au plus lointain voisin (cette méthode reprend la mesure de Casertano & Hut, 1985). On observe que cette évaluation, tracée par le nuage de points rouge, est très proche de l'estimateur de HOP. Le profil théorique du modèle de King est tracé en noir pour référence. Comme attendu, l'augmentation de la valeur du paramètre N_{dens} conduit à des distributions moins dispersées ce qui permet à la méthode de mieux ajuster le profil théorique. En revanche, pour une valeur de $N_{\text{dens}} = 64$ correspondant à la valeur par défaut de la méthode, on observe que l'estimation aux bords de l'amas est surestimée. En effet, la densité du modèle de King chute brutalement au niveau du rayon de troncature. Ainsi, les zones de faibles densités où il n'y a que peu voire pas de particules ne sont pas correctement estimées. Cette effet est donc aggravé pour des configurations n'ayant pas un grand nombre d'étoiles comme l'amas présenté ici.

Identification de groupes

HOP se base sur la densité locale des particules pour les regrouper en groupes. Elle consiste à regrouper les particules en fonction de leur proximité relative à un pic local de densité. Plus précisément, la procédure consiste à rechercher dans les N_{HOP} voisins la particule la plus dense de manière itérative, jusqu'à converger vers une particule n'ayant pas de voisin plus dense qu'elle même, soit un maximum local. La valeur du paramètre N_{HOP} règle l'aspect local de cette recherche. Ce paramètre n'est pas de première importance et l'on choisira, dans un souci de cohérence, une valeur inférieur ou égale à N_{dens} . Cette procédure est rapide et robuste car elle permet une recherche de maximum en suivant les gradients de densité de la distribution, sans avoir à les calculer. De cette manière HOP classe l'ensemble des particules autour des pics de densité locaux comme on peut le voir sur le panneau de gauche de la Fig. 3-4. Ce classement permet ensuite l'identification de groupes à partir de 3 seuils.

- Le paramètre principal correspond au seuil à partir duquel une particule est considérée comme faisant partie d'une surdensité. On définit une densité minimale δ_{outer} pour qu'une particule fasse partie d'un groupe. Toutes les particules ayant une densité en dessous de cette valeur sont considérées comme faisant partie du champ (voir le panneau de droite de la Fig. 3-4).

Deux autres seuils permettent d'affiner la sélection des groupes finaux. En effet, on peut se désintéresser des

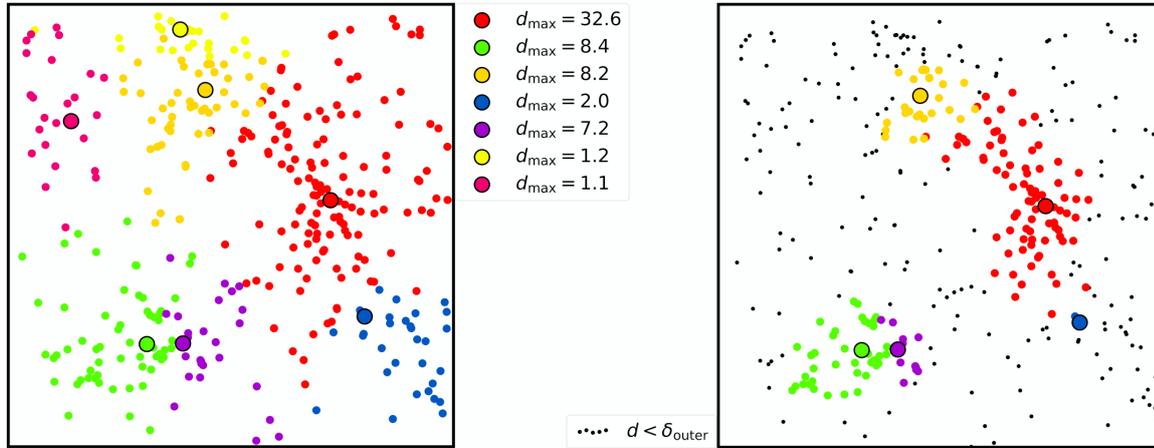


FIGURE 3.4 – Procédure de classement d'une distribution de particule par HOP. Panneau de gauche : Chaque particule est regroupée autour d'un maximum local de densité. Chaque couleur correspond à un regroupement autour des particules les plus denses localement (marquées par un cercle noir ; leur valeur dans la légende est exprimée en fraction de la densité moyenne). Panneau de droite : seules les particules dont la densité excède une valeur seuil δ_{outer} sont conservées (ici $\delta_{outer} = 2$).

régions dont le pic de densité n'est pas assez important ou lorsque de multiples pics sont trop proches les uns des autres.

- Le seuil δ_{peak} , consiste à évaluer si le pic de densité local est assez significatif. Sur la Fig. 3.5, cela consiste à éliminer le groupe bleu de la Fig. 3.4.
- δ_{saddle} permet de fusionner différents groupes dont la densité à leur frontière ne descend pas en dessous d'une densité de selle. Ainsi sur la Fig. 3.5, cela amène les groupes bleu et vert à fusionner.

Ces trois seuils, que l'on choisira de manière à ce que $\delta_{outer} < \delta_{saddle} < \delta_{peak}$, correspondent aux différents contours en densité permettant la définition précise des groupes. δ_{outer} définit le contour entre les groupes et le champ, δ_{saddle} celui entre deux groupes mitoyens, et δ_{peak} celui définissant la valeur minimale de densité à atteindre pour être considéré comme significatif.

Ils permettent une grande modularité dans l'identification de différentes régions d'intérêt. Cela constitue à la fois la force et la faiblesse de la méthode car la définition des groupes est donc sujette à leur choix. Néanmoins la méthode de classification de l'ensemble des particules est robuste. Ainsi, au delà d'identifier seulement les pics de densités d'une distribution, cette méthode peut aussi être adaptée pour sélectionner des ensembles plus larges autour de ces derniers. Cela nous a notamment permis d'extraire des sous-ensembles cohérents des modèles GDF. La Fig. 3.6 montre une sous-sélection d'un modèle GDF réalisée avec l'algorithme HOP. Cela permet d'isoler les structures plus étendues et moins denses (comme les filaments) autour des pics de densité et avec lesquelles elles sont en interaction.

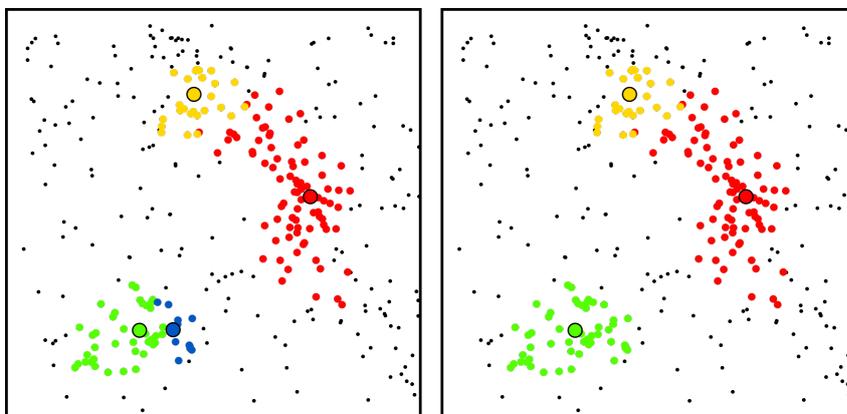


FIGURE 3.5 – Procédure d'affinage des groupes par HOP. Panneau de gauche : seuls les groupes dont le pic de densité et significatif sont conservés. Panneau de droite : les groupes dont les pics de densités sont suffisamment proches sont scindés.

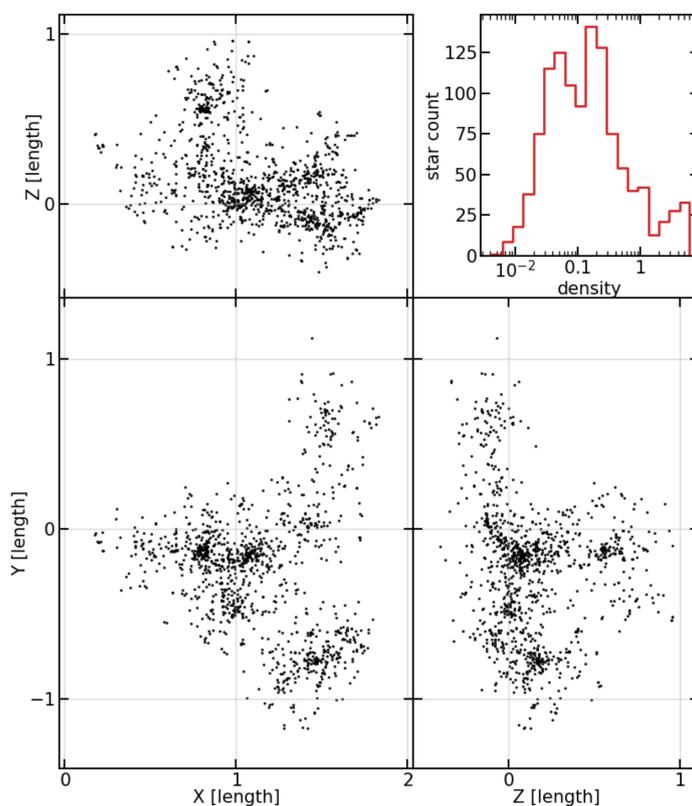


FIGURE 3.6 – Ensemble de 3 cartes de projections orthogonales d'une sous-sélection de plusieurs pics de densités et de leur environnement réalisée grâce à l'algorithme HOP. Le panneau en haut à droite indique la distribution de densités des particules sélectionnées et couvrant trois ordres de grandeurs.

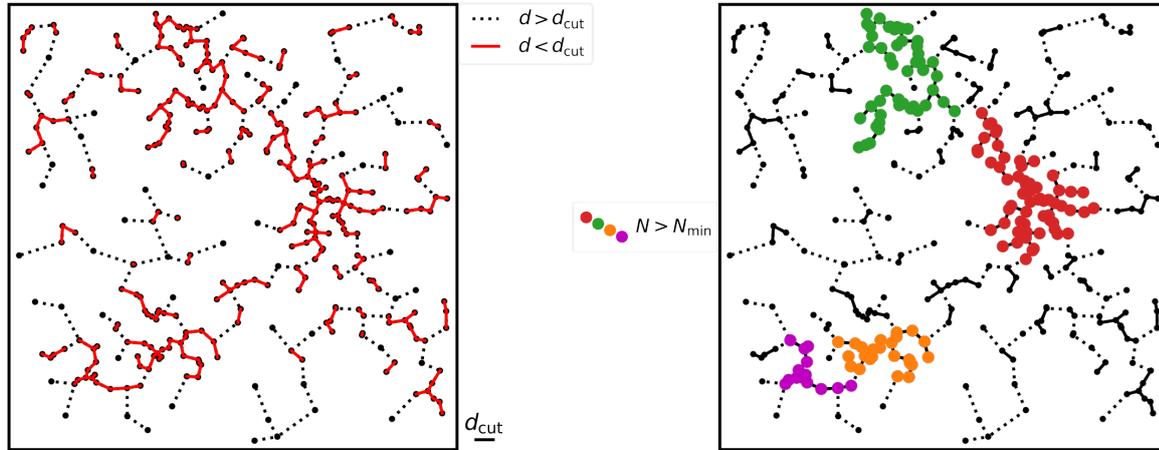


FIGURE 3.7 – Procédure d'identification de groupes par la méthode du MST. Panneau de gauche : les pointillés tracent le MST sur l'ensemble de la distribution. Les traits pleins rouges, représentent le graphe une fois qu'il a été « coupé » pour des distances supérieures à d_{cut} , symbolisée en bas de la figure entre les deux panneaux. Panneau de droite : Seuls les groupes comportant un nombre de membre supérieur à N_{min} sont considérés comme significatifs.

3.2.2 L'algorithme MST

L'algorithme HOP est un outil puissant pour isoler différentes structures des modèles GDF. Néanmoins son application n'est que peu répandue dans les observations. Ainsi, dans le but d'établir un parallèle entre les modèles GDF et les observations, nous nous sommes intéressés à l'algorithme du MST. Le Minimum Spanning Tree a en effet beaucoup été utilisé dans les études de régions de formation d'étoiles pour identifier des surdensités stellaires au sein de régions jeunes (Gutermuth et al., 2009; Kirk & Myers, 2011) ou les nuages denses du MIS (Billot et al., 2011; Beuret et al., 2017). Il a aussi servi comme outil pour distinguer différents degrés de fragmentation au sein des amas (Cartwright & Whitworth, 2004) ou comme mesure de leur ségrégation en masse (Allison et al., 2009)

Le MST vient de la théorie des graphes. Il consiste à lier les points d'une distribution sans former de boucle fermée tout en minimisant la distance totale du graphe ainsi tracé. Le graphe obtenu est constitué d'un ensemble d'arêtes représentant la distance aux plus proches voisins de chaque particule. Il est ainsi possible d'identifier des surdensités en scindant le graphe lorsque la longueur d'une arête est supérieure à une valeur seuil d_{cut} . On peut visualiser cette procédure sur le panneau de gauche de la Fig. 3.7. Les particules sont ainsi isolées en îlots. La deuxième étape consiste à ne conserver que les îlots possédant un nombre significatif de membres N_{min} , comme on peut le voir sur le panneau de droite de la Fig. 3.7

Le graphe tracé par le MST suit par construction, l'agencement hiérarchique de la distribution et n'impose pas de forme particulière aux surdensités isolées. Cela lui permet d'identifier une distribution dense de points alignés, le long d'un filament par exemple. Cette propriété est particulièrement intéressante pour l'étude des structures au sein des régions de formation d'étoiles.

La procédure du MST est efficace et ne requiert que deux paramètres à contraindre. Le premier, qui est

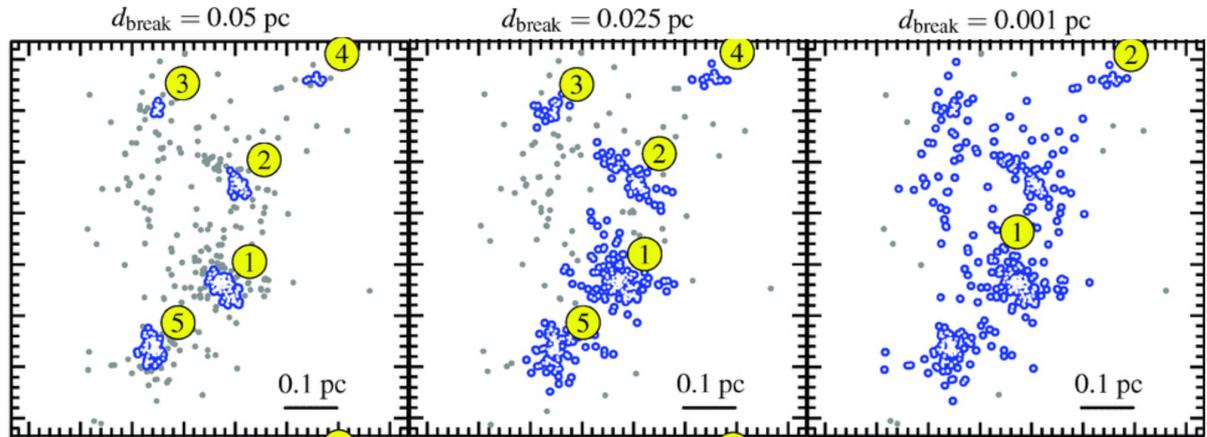


FIGURE 3.8 – Identification de sous-amas par la méthode du MST pour différentes valeurs de la longueur de coupure (issue de [Maschberger et al., 2010](#)). La valeur de d_{cut} retenue correspond au panneau du milieu.

la distance de coupure d_{cut} , correspond au critère pour isoler les particules en moyenne plus proches les unes que les autres que le reste de l'échantillon. Ce paramètre permet de faire une distinction simple entre des environnements denses et plus dispersés. Néanmoins une détermination absolue de sa valeur paraît difficile car elle dépend fortement de la distribution étudiée. Dans certaines situations, une simple inspection visuelle est suffisante. Par exemple, c'est l'approche adoptée par [Maschberger et al. \(2010\)](#) lors d'une analyse de la distribution de proto-étoiles issues de simulations hydrodynamiques de [Bonnell et al. \(2003, 2008\)](#). Les étoiles créées par les simulations se concentrent en sous-amas bien identifiables. Ainsi, la valeur de d_{cut} permettant de les isoler s'impose naturellement (voir Fig. 3.8).

Cependant cette distinction n'est pas forcément aussi évidente. Pour les amas GDF de plusieurs milliers de particules, nous avons été amenés à employer une méthode plus systématique. Une technique couramment utilisée est celle proposée par [Gutermuth et al. \(2009\)](#). Elle consiste à exploiter la distribution de l'ensemble des arêtes tracées par le MST. En pratique, la distribution cumulative des arêtes est ajustée par deux régressions linéaires, l'une aux petites échelles correspondant au régime où les particules sont groupées et l'autre à plus grande échelle correspondant à un régime plus diffus. L'intersection de ces deux droites correspond à la valeur séparant les deux environnements et représente ainsi la valeur d_{cut} choisie. La Fig. 3.9 illustre cette méthode.

Une méthode alternative, consiste à identifier les deux régimes en comptant le nombre de groupes identifiés lorsque la valeur de la longueur d_{cut} varie, pour une valeur N_{min} fixée. Lorsque d_{cut} est proche de la distance minimale entre deux particules, on isole seulement quelques rares groupes très concentrés. En augmentant d_{cut} , le nombre de groupe croît, ainsi que le nombre de membres au sein des groupes déjà identifiés. Cette tendance se poursuit jusqu'à ce que d_{cut} dépasse une distance caractéristique entre deux membres de groupes différents. À ce moment là, les groupes précédemment identifiés fusionnent et le nombre de groupes total se met à chuter. Cette distance caractéristique maximisant le nombre de groupes identifiés représente une valeur adéquate pour le choix de d_{cut} . On notera que, contrairement à la méthode de [Gutermuth et al. \(2009\)](#), cette méthode requiert de définir préalablement un nombre minimum de membre par groupe, la valeur N_{min} . Ce paramètre est dans la plupart des études fixé à $N_{\text{min}} \sim 12$ ([Billot et al., 2011](#); [Maschberger et al.,](#)

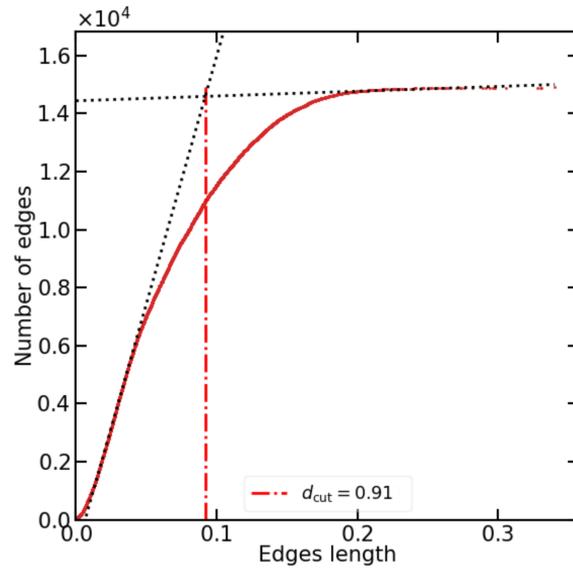


FIGURE 3.9 – Détermination de la longueur de coupure selon la méthode de Gutermuth et al. (2009) via la distribution cumulative des arêtes du MST. Les deux régressions linéaires correspondent aux régimes groupé et dispersé (ligne pointillée noir). L'intersection de ces deux droites indique la valeur de d_{cut} .

2010). Cette valeur est considérée comme un bon compromis dans le but de déceler de petits groupes sans les confondre avec des systèmes multiples. Néanmoins elle peut être considérée comme faible dans des environnements plus denses. Par exemple Lada & Lada (2003) considèrent un groupe d'étoiles comme significatif s'il contient au minimum 35 membres. La Fig. 3.10 montre l'histogramme du nombre de groupes identifiés en fonction de la valeur de d_{cut} , pour différente valeur de N_{min} . On remarque que la valeur de d_{cut} maximisant le nombre de groupes reste similaire dans chacun des cas. On conclut ainsi que le nombre minimum de membres par groupe influe peu sur la détermination de la distance de coupure.

Comme l'avait déjà observé Dorval et al. (2016), les deux méthodes présentées ci-dessus conduisent à des valeurs de d_{cut} très proches. Il faut toutefois garder à l'esprit que les valeurs de d_{cut} et de N_{min} peuvent être sujettes à de petites variations. Ainsi, il pourra être opportun d'adapter le choix de ces valeurs à l'échantillon étudié ou aux surdensités à identifier.

Dans le cas de données issues d'observations réelles, l'identification de surdensités d'étoiles est toujours un exercice délicat. Dans ce contexte, leur calibration sur des modèles théoriques comme les modèles GDF reste un bon moyen dans le but de lever les biais auxquels ils peuvent être soumis. Un de ces biais est étudié directement au chapitre 4 suivant.

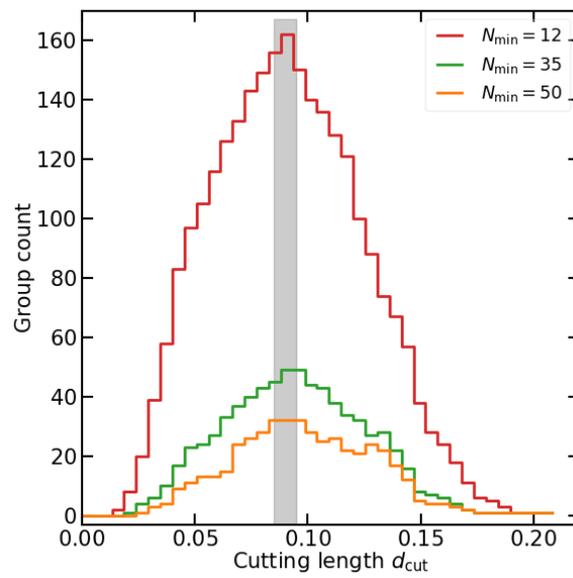


FIGURE 3.10 – Histogramme du nombre de groupes en fonction de la longueur de coupe d_{cut} , pour différentes valeurs de N_{min} . La valeur de d_{cut} maximisant le nombre de groupes vaut 0.9 ± 0.05 .

BIBLIOGRAPHIE

- Allison, R. J., Goodwin, S. P., Parker, R. J., et al. 2009, *MNRAS*, 395, 1449
- Bastian, N., Covey, K. R., & Meyer, M. R. 2010, *ARA&A*, 48, 339
- Bate, M. R. 2019, *MNRAS*, 484, 2341
- Bate, M. R., Bonnell, I. A., & Price, N. M. 1995, *MNRAS*, 277, 362
- Beuret, M., Billot, N., Cambr esy, L., et al. 2017, *A&A*, 597, A114
- Billot, N., Schisano, E., Pestalozzi, M., et al. 2011, *ApJ*, 735, 28
- Bonnell, I. A., Bate, M. R., & Vine, S. G. 2003, *MNRAS*, 343, 413
- Bonnell, I. A., Clark, P., & Bate, M. R. 2008, *MNRAS*, 389, 1556
- Cartwright, A. & Whitworth, A. P. 2004, *MNRAS*, 348, 589
- Casertano, S. & Hut, P. 1985, *ApJ*, 298, 80
- Chabrier, G. 2003, *PASP*, 115, 763
- Dorval, J., Boily, C. M., Moraux, E., Maschberger, T., & Becker, C. 2016, *MNRAS*, 459, 1213
- Eisenstein, D. J. & Hut, P. 1998, *ApJ*, 498, 137
- Federrath, C., Banerjee, R., Clark, P. C., & Klessen, R. S. 2010, *ApJ*, 713, 269
- Gouliermis, D. A. 2018, *PASP*, 130, 072001
- Gutermuth, R. A., Megeath, S. T., Myers, P. C., et al. 2009, *ApJS*, 184, 18
- King, I. 1962, *AJ*, 67, 471
- Kirk, H. & Myers, P. C. 2011, *ApJ*, 727, 64
- Kroupa, P. 2001, *MNRAS*, 322, 231
- Krumholz, M. R. 2014, *Phys. Rep.*, 539, 49
- Lada, C. J. & Lada, E. A. 2003, *ARA&A*, 41, 57
- Maschberger, T. 2013, *MNRAS*, 429, 1725

Maschberger, T., Clarke, C. J., Bonnell, I. A., & Kroupa, P. 2010, *MNRAS*, 404, 1061

Motte, F., Bontemps, S., & Louvet, F. 2018, *ARA&A*, 56, 41

Salpeter, E. E. 1955, *ApJ*, 121, 161

EFFETS DE PROJECTION DANS LES RÉGIONS DE FORMATION STELLAIRE [ARTICLE]

La structure fragmentée des régions de formation d'étoiles rend leur analyse complexe en raison des nombreuses asymétries qu'elles présentent. En effet, l'étude de ce type de configuration est un enjeu crucial dans le but de contraindre précisément les processus qui la modèlent. Les observations se heurtent néanmoins à de nombreuses limitations dont l'imprécision de la distance relative des étoiles le long de la ligne de visée est l'une des plus criantes. À l'inverse, il est possible à travers les simulations de connaître l'ensemble de l'espace des phases d'une région. Dans ce chapitre, nous nous intéresserons donc aux différences dans l'identification de surdensités stellaires entre simulations et observations. Nous estimerons le biais observationnel due à une identification de groupes d'étoiles en projection comparée à une identification quasi-idéale 3D. Cette étude a fait l'objet d'un article dans le journal *Astronomy & Astrophysics* en octobre 2020 que nous reproduisons ci-après.

En utilisant des modèles de fragmentation à N -corps (GDF), permettant de reproduire les distributions fragmentées d'amas jeunes d'étoiles, nous avons étudié la différence entre des groupes identifiés en 2D et en 3D par l'algorithme du MST. Dans le but de quantifier le biais observationnel induit par cette identification incomplète, nous avons estimé cette différence sur les propriétés de ces groupes. Nous avons en particulier mesuré leur masse dynamique car elle permet aussi d'évaluer leur équilibre dynamique. Les principaux résultats de cette étude sont listés ci-dessous :

- La masse dynamique des groupes identifiés en projection est systématiquement sous-estimée de 50 % dans la moitié des cas et atteint 100 % dans le quart d'entre eux.
- Les effets de projection conduisent à l'identification de groupes ayant en moyenne un rayon plus petit de 15 à 30% et possédant une dispersion des vitesses plus faibles de 20 à 30%
- Nous avons par ailleurs montré que plus l'incomplétude des observations augmentait, plus cette erreur systématique était marquée.
- De plus, nous avons estimé que les dernières données *Gaia* DR3 ne permettrait pas de limiter cet effet systématique au sein de régions de taille inférieure à la dizaine de parsecs.
- Enfin, bien que les groupes identifiés en 3D par le MST ne soient pas relaxés et toujours en interaction entre eux, on observe bien une corrélation entre leur masse réelle et leur masse dynamique. La valeur du facteur de proportionnalité semble néanmoins être inférieur à celle typiquement utilisé. Nous reviendrons plus en détail sur cette particularité au chapitre suivant.

Assessing membership projection errors in star forming regions

T. Roland, C. M. Boily, and L. Cambrésy

Université de Strasbourg, CNRS, Observatoire Astronomique de Strasbourg, UMR 7550, 67000 Strasbourg, France
e-mail: timothe.roland@astro.unistra.fr

Received 6 August 2020 / Accepted 19 October 2020

ABSTRACT

Context. Young stellar clusters harbour complex spatial structures emerging from the star formation process. Identifying stellar over-densities is a key step in better constraining how these structures are formed. The high accuracy of distances derived from *Gaia* DR2 parallaxes still do not allow us to locate individual stars within clusters of ≈ 1 pc in size with certainty.

Aims. In this work, we explore how such uncertainty on distance estimates can lead to the misidentification of membership of sub-clusters selected by the minimum spanning tree (MST) algorithm. Our goal is to assess how this impacts their estimated properties.

Methods. Using N -body simulations, we build gravity-driven fragmentation models that self-consistently reproduce the early stellar configurations of a star forming region. Stellar groups are then identified both in two and three dimensions by the MST algorithm, representing respectively an inaccurate and an ideal identification. We compare the properties derived for these resulting groups in order to assess the systematic bias introduced by projection and incompleteness.

Results. We show that in such fragmented configurations, the dynamical mass of groups identified in projection is systematically underestimated compared to those of groups identified in 3D. This systematic error is statistically of 50% for more than half of the groups and reaches 100% in a quarter of them. Adding incompleteness further increases this bias.

Conclusions. These results challenge our ability to accurately identify sub-clusters in most nearby star forming regions where distance estimate uncertainties are comparable to the size of the region. New clump-finding methods need to tackle this issue in order to better define the dynamical state of these substructures.

Key words. methods: statistical – stars: formation – stars: kinematics and dynamics – open clusters and associations: general

1. Introduction

Young stars are predominantly observed in the dense parts of spiral arms of disc galaxies. After a decade of *Herschel* observations, it is now established that these star forming regions exhibit highly sub-structured configurations due to fragmentation of the gas in extended filaments (André et al. 2010; Pokhrel et al. 2018). The latest interferometric facilities (e.g., ALMA, NOEMA) allow to study filaments and primordial cores at sub-parsec scales (see e.g., Hacar et al. 2017, 2018; André et al. 2019; Montillaud et al. 2019), bringing to light the very last stage of the transformation of gas into stars. Stars emerge from this network in groups of varying size and morphology. However, their kinematics is at odds with that of their natal gas (Foster et al. 2015). Moreover, their dynamical state is poorly constrained (Kuhn et al. 2019) and their fate uncertain.

The new era of precise astrometry opened by *Gaia* DR2 enables exploration of the spatial and kinematic properties of a huge number of young stellar systems (see e.g., Kounkel & Covey 2019; Cantat-Gaudin et al. 2019 for the newest catalogs of *Gaia*-discovered stellar clusters). In the solar neighbourhood, the 3D structure of local star forming clouds is within our reach. Großschedl et al. (2018) measured the extended 3D structures of the Orion A tail, reporting an estimated total length of 90 pc, in significant disagreement with the projected size of 40 pc. Other large, extended structures have been identified outside the solar circle (e.g., Alves et al. 2020). The astonishing accuracy of *Gaia* DR2 data peaks at $M_G \approx 13$, but degrades steadily at fainter magnitudes. This leads to uncertainties in distance estimates affecting mostly low-luminosity or reddened stars. In young and embedded regions, identifying 3D

stellar structures at (sub-)parsec scales remains challenging. For instance, the selected sample of Kounkel et al. (2018) in Orion has a mean uncertainty on parallax measurements of ≈ 0.5 mas which corresponds to ≈ 8 pc at this distance (≈ 400 pc).

In this context, simulations are key to testing our understanding of the processes acting in these fragmented regions. Recent computations of hydro-gravitational fragmentation and accretion, coupled with radiation-transfer physics, give promising results in terms of for example the shape of the stellar mass function (e.g., Bate 2012), formation of proto-planetary discs (e.g., Hennebelle et al. 2016), and the role of stellar feedback (e.g., Wall et al. 2020). Despite continued progress, the number of young stars produced in these high-resolution calculations is limited to a few hundred. Larger calculations remain challenging. Moreover, strong interactions between newly born stars occur in these simulations. Exploring the interplay between several of these systems is key to probing the formation of more massive clusters. Consequently, pure N -body modelling is still of great use to explore configurations with $N > 1000$ stars. This approach works well when gravity takes over magneto-thermal support and drives the fragmentation of the giant molecular cloud (GMC). This allows us to address specific topics such as primordial mass segregation (Pavlík et al. 2019) or the impact of gradual star formation (Farias et al. 2019). As massive clusters are also likely to form massive stars (Motte et al. 2018), N -body simulations are also suited to exploring the environment where they form.

There is a long-standing history of modelling stellar interactions with gravity alone. One major challenge of such work consists in generating realistic initial configurations. Different methods have been developed. One of the most straightforward is to follow snapshots of hydro-dynamical simulations and

the structures traced by sink particles (e.g., Moeckel & Bate 2010; Fujii & Portegies Zwart 2016). This approach is well suited to exploring the long-term evolution of the original hydro-dynamical simulations, but is limited by the high computational effort to model the full star formation process. An efficient alternative that can be used to reproduce the fragmented aspect of star forming complexes at lower cost was introduced by Goodwin & Whitworth (2004), namely the box fractal method. This technique can recreate spatial configurations of various observed clusters and allows the user to tune the different levels of substructures. However, the velocity field is not constrained by the history of formation of such structures which makes the dynamical state of the system somewhat ad hoc.

Another method providing a balance between computational cost and realistic configurations was proposed by Dorval et al. (2016), which we refer to as gravity-driven fragmentation (GDF) models. This method is an efficient way to build fragmented initial conditions self-consistently using only N -body modelling. It allows the user to explore a large number of fragmented configurations and test our understanding of the dynamical processes in action (see also Dorval et al. 2017).

In this paper, we specifically discuss morphological biases introduced when estimating properties of stellar over-densities in a fragmented cluster seen in projection. We used the GDF method to model a realistic distribution of stars in a young star cluster (Sect. 2). Our methodology consists in identifying stellar groups of more than 50 members (also referred to as clumps), both in 3D and in projections (2D) and estimating their properties (Sect. 3). We used the virial dynamical mass to compare groups identified by both methods and assess the deviation introduced by projection effect (Sect. 4). Finally, we discuss the implications of our results in Sect. 5 and conclude in Sect. 6.

2. Building realistic stellar distribution

2.1. Gravity-driven fragmentation modelling

We start our study using the GDF models described in Dorval et al. (2016). We summarise here the motivations and principles of such models.

The fragmentation of a continuous cold fluid bound by gravity will proceed from continued growth of a spectrum of perturbations. When dealing with stars, the point-mass nature and limited number of elements means that the potential is not smooth, and Poisson-seeded density fluctuations rapidly develop into dense substructures. The growth of small-scale modes will proceed on very short timescales, however large-scale modes will grow on the global gravitational timescale of $\sim 1/\sqrt{G\rho_0}$ which is the same as the free-fall timescale. To allow time for these fluctuations to develop well in the non-linear regime, one sets up a self-gravitating system uniformly populated by stars with an outward radial velocity field. The velocity flow is taken to be isotropic with respect to the barycentre of the system; its amplitude is such that the total binding energy remains negative, and so expansion stops after some time which we take to coincide with the formation timescale of massive stars.

This method allows us to simulate the two main physical processes driving the late stages of star formation: self-gravity, computed by a collisional N -body integrator; and adiabatic cooling of the system, driven by the expansion. We adopted such a modelling to reproduce the distribution of young stars in a filamentary network whose nodes are dense and asymmetric clumps. These clumps comprise between tens of stars and a few hundred, and are usually considered as the early prod-

ucts of the star formation process (Krumholz 2014). As the overall velocity field is self-consistently built, we are able to probe their dynamical state and interactions with their surroundings. Furthermore, when using a mass spectrum, the most massive stars attract the surrounding low-mass stars to themselves. Dorval et al. (2016) showed that massive stars are the seeds for the growth of fragment modes. As a result, clumps are mass segregated at birth, retrieving a key feature of hydro-simulations (e.g., Maschberger & Clarke 2011; Wall et al. 2019). They also harbour a top-heavy mass profile whose Salpeter index is comparable to the one found in young- and open clusters of the Milky Way (Bastian et al. 2010). Figure 1 shows the initial, intermediate, and final steps for one of our models. More details and further examples are given in Dorval et al. (2016, 2017).

2.2. Simulation details

We based our study on a GDF model with $N = 15\,000$ young stars following a canonical Kroupa-Chabrier initial mass function (IMF; Kroupa et al. 2001; Chabrier 2003). We adopt the functional form of Maschberger (2013) for efficiency. The initial velocity field is chosen so that the expansion stops roughly when the system is 4 pc in diameter, resulting in a mean density of

$$\rho_0 \approx 150 M_\odot \text{pc}^{-3}.$$

These parameters probe the high-mass regime of young clusters close to the so-called young massive clusters (YMCs). They were chosen for several reasons.

– Firstly, these parameters allow us to accurately sample the IMF in the range $0.01\text{--}30 M_\odot$. The high-mass end limit was chosen in order to keep the mean mass of the system around $\approx 0.3 M_\odot$. Indeed a specific draw of the IMF on 15 000 stars leads to approximately ten stars more massive than $30 M_\odot$ ¹ and can contribute as much as 10% of the mass budget of the overall region.

– Secondly, these allow us to reproduce a well-defined sample of clumps of different memberships and densities. This is a direct consequence of including enough massive stars that act as the seeds of the formation of these clumps. In a theoretical “bottom up” scenario of star formation, where these clumps will subsequently merge together, such a system is a likely progenitor of a massive open cluster. This large range in densities (see Sect. 2.3) allows us to probe different clump environments. However, we note that such a crowded configuration can affect the level of contamination. In Sect. 5.2, we discuss a model with lower membership.

– Finally, with such parameters, the formation of the model lasts 0.55 Myr which matches the typical time for massive stars ($>1 M_\odot$) to reach the main sequence. This also corresponds to the free-fall time of a GMC with density equal to ρ_0 ,

$$t_{\text{ff}} = \sqrt{\frac{3\pi}{32G\rho_0}} = 0.67 \text{ Myr}. \quad (1)$$

As most of the stars in a GMC are formed within a free-fall time (Grudić et al. 2018), this allows us to study the very early stages of the configuration of newly born stars. These structures have not yet been erased by two-body relaxation, and their dynamical state is a direct result of the fragmentation process.

We perform the computations with the AMUSE platform (Portegies Zwart & McMillan 2018). We used the N -body code

¹ The probability to draw a star more massive than $30 M_\odot$ is of the order of 10^{-4} with $N \sim 10^4$.

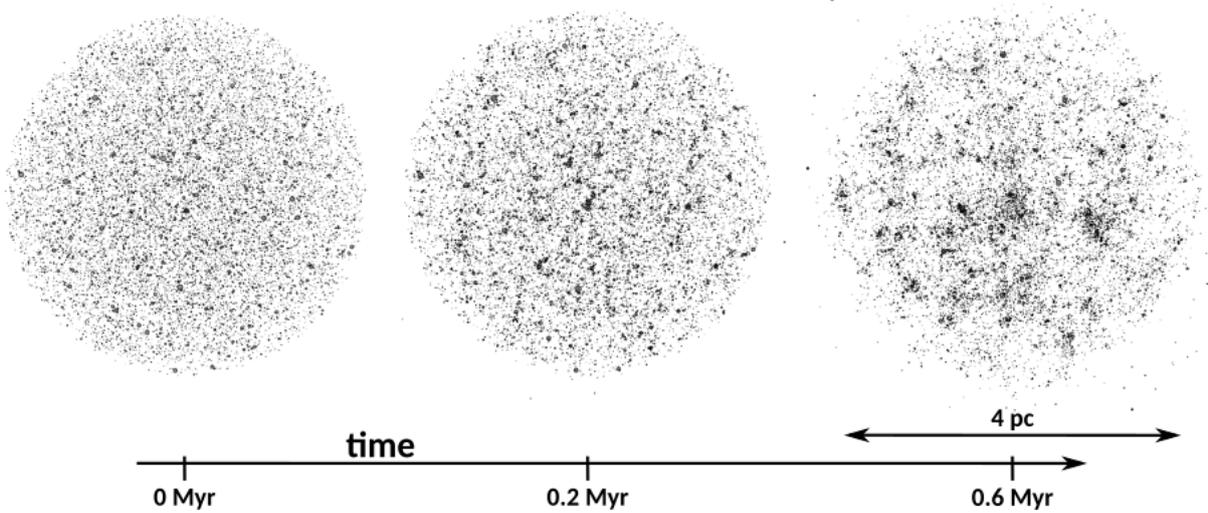


Fig. 1. Gradual fragmentation of a GDF model in co-moving coordinates (physical length scale is only valid for the *right panel*). *Left panel*: initial step: stars are uniformly distributed and have radial velocities. *Middle panel*: growth of fluctuations in the expanding sphere. *Right panel*: point at which the system is fully fragmented and expansion stops.

ph4 which is a fourth-order Hermite scheme integrator coupled with the `Multiple` module to handle efficiently close interactions (Hut et al. 1995; McMillan & Hut 1996). The relative energy drift $\delta E/E$ at each time-step is always lower than 10^{-6} except when close encounters cannot be resolved by the `Multiple` module, and can lead to an error up to 10^{-4} . However, these remain rare, which makes the total energy drift during the whole calculation of the order of 10^{-4} . We do not explicitly include any binary population but they are able to form dynamically through the fragmentation process. However, the fraction of multiple stars remains below 5% and these do not influence our conclusions.

2.3. Sub-selection within the GDF model

Gravity-driven fragmentation models allow exploration of the structures built by the fragmentation process. In this study, we specifically target clumps that can be misidentified when only seen in projection. However, as we see in Fig. 1, the spherical imprints of the initial conditions remain visible at the large scale. This overall geometry of the cluster can have an impact on the clumps that we identify in projection in the following section.

To this end, we implemented a method to extract a subsample from the central parts of the system. This method is based on the HOP algorithm (Eisenstein & Hut 1998) which was initially designed to identify over-dense haloes in any particle-based sample. HOP assigns to each particle a local density estimate based on a list of nearest neighbours and groups them around density peaks. We adapted the HOP algorithm to select a sample representative of the complex structure of the GDF model spanning a large-enough range in density. First, we tune the parameters to select only particles with a local density above a given threshold. Several tests lead us to set the threshold value equal to $\approx 75 M_{\odot} \text{pc}^{-3}$, or half of the median system density. We then exclude groups that were close to the edge of the GDF sphere by retaining only those that included at least one member lying less than 1 pc from the centre of the model. This two-step procedure leads to the selection of a subsample of particles excluding about half of the stars.

This new sample is the reference set for the rest of the study. Figure 2 shows three orthogonal views of the initial GDF

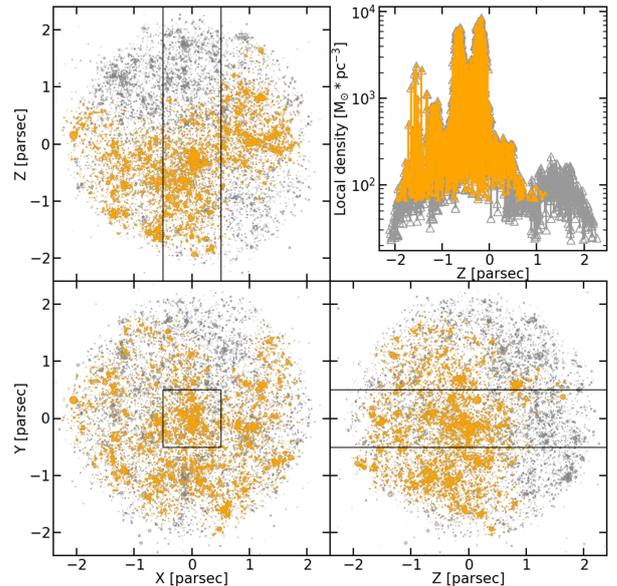


Fig. 2. Three orthogonal views of the GDF model. The full initial sample of stars is represented in grey shades and the sub-sample we extracted is shown in orange. *Upper right panel*: local density of the particles estimated by HOP stacked along the slice represented by the black rectangles on the three other views.

model (grey dots) and the selection of 7235 stars (orange dots). On the upper left panel, local densities of particles within a slice of the model (black rectangles) are plotted. This highlights that the selection retains all the main density peaks. Furthermore, local densities span a large range, from $\approx 100 M_{\odot} \text{pc}^{-3}$ to $\approx 10^4 M_{\odot} \text{pc}^{-3}$.

3. Stellar groups

Stars are more likely to be formed in dense environments where the final collapse of gas is transformed into stars through an accretion process. The various stages of accretion² means that

² These are known as Class 0, I, II, III objects (see e.g., Andre et al. 2000).

stars acquire their mass in a shared environment. Data from *Herschel* for example have long indicated that the proximity of young stars gives rise to extended structures of varying morphology, all interconnected by a complex network of bridges and filaments. We are tasked with identifying the stars that belong to such a network.

3.1. Identification method

Several techniques have been developed to detect over-densities in observations of star forming regions. These include for example, the minimum spanning tree (MST) algorithm (Gutermuth et al. 2009); the finite-mixture model technique (Kuhn et al. 2014); DBSCAN (Joncour et al. 2018), and HDBSCAN (Kounkel et al. 2018). Despite their specific features, most of these techniques rely on star number density estimates projected on the sky³. This is mainly because of the few observables available in such crowded regions, where extinction may be significant. By construction, these methods may miss the true underlying 3D structure of the objects under study.

We address the biases that arise from projection effects, focusing on the MST method. Its ability to detect over-dense structures of any shape makes it a powerful tool in studies of star forming regions. It has been widely used by both theorists (e.g., Maschberger et al. 2010) and observers (e.g., Kirk & Myers 2011; Beuret et al. 2017). MST statistics can also be used to derive other properties of stellar clusters such as morphological aspects (Cartwright & Whitworth 2004) or a measure of mass segregation (Allison et al. 2009). Moreover, the MST method may be adapted to treat either 2D or 3D space distributions. Our setup described below was similarly implemented in both cases. This uniformity in the methodology is crucial in order to avoid introducing additional biases between 2D and 3D identifications.

3.2. Minimum spanning tree setup

We summarise the MST procedure briefly (see e.g., Gutermuth et al. 2009 for further details). A spanning tree is a path linking all particles of a data set without forming closed loops. The MST is the spanning tree that minimises the total length of this path. We thus obtain a hierarchical set of edges that link every particle of the sample. To identify groups, we define a characteristic cut-off scale, d_{cut} , such that all edges of larger values will be removed from the MST set. This isolates star aggregates that are on average closer to each other. The choice of d_{cut} is critical to our analysis: a large value will link together stars that are very widely separated, revealing a few populous and sub-structured groups of stars; on the other hand, a very small value will isolate binary and multiple stars, but will not reveal the hierarchical complex structure of the system as a whole. A natural way to pick a value for d_{cut} is to count the number S of subsets with a minimum membership, and vary d_{cut} . When decreasing d_{cut} from $\approx R$, the size of the system, it is not too hard to see that S should rise from a value of 1 to a maximum value S_{max} and drop to ≈ 0 when $d_{\text{cut}} \rightarrow 0$. An optimal value of d_{cut} is found when maximising the number of groups, $S = S_{\text{max}}$ (see Dorval et al. 2016). However, this iterative procedure can prove computationally costly. After some experimentation, we settled on a simple but efficient procedure based on the distribution of the MST edges. A characteristic

length is retrieved by adding one-half of a standard deviation to the mean value of MST edges. The cutting length d_{cut} is set to

$$d_{\text{cut}} = \text{mean}(E_{\text{MST}}) + \text{std}(E_{\text{MST}})/2, \quad (2)$$

with E_{MST} being the set of MST edges. This way of fixing d_{cut} provides nearly optimal values without the iteration procedure.

3.3. Membership N

We fixed a minimum number of stars per stellar group, N , based on two arguments. The first is that we consider a system that is on the order of 1 Myr old, which is old enough for massive stars to have reached the main sequence, but not so old that the system as a whole has had time to settle into virial equilibrium. Hence, filaments and knots are transient features. On the other hand, we want the majority of these clumps to be as close as possible to their state at formation: indeed, MST statistics cannot be used to gauge the equilibrium state of the substructures we aim to isolate. However, what is clear is that any substructure will evolve internally by phase mixing and kinetic-energy diffusion on a timescale of $\propto 1/\sqrt{G\rho}$. We take as reference an average density of $100\gamma M_{\odot} \text{pc}^{-3}$, with γ being a dimensionless scalar. We then compute a characteristic time $t_{\text{cr}} \sim 1.5/\sqrt{\gamma} \times 10^6$ yr, which we take as the dynamical timescale for motion driven by gravity. When a number N of stars are bound together, their repeated interactions lead to diffusion of kinetic energy on a timescale t_{rel} given by (e.g., Meylan & Heggie 1997):

$$t_{\text{rel}} \simeq 0.138 \left(\frac{R_{\text{h}}}{2R_{\text{v}}} \right)^{\frac{3}{2}} \frac{N}{\ln 0.4N} t_{\text{cr}}. \quad (3)$$

In the above relation, the ratio of half-mass radius R_{h} to virial radius $R_{\text{v}} = -GM/E$ (where E is the binding energy) is ≈ 1 for self-gravitating configurations near equilibrium. We find $t_{\text{rel}} > t_{\text{cr}}$ for $N \gtrsim 50$, and furthermore $t_{\text{rel}} > 1$ Myr for $\gamma \leq 1$. This tells us that clumps with membership N of 50 or more should remain relatively close to their phase-space configuration at birth. Even so, the case for setting $N = 50$ becomes weaker in denser environments ($\gamma \gg 1$), and we should increase N in proportion to $\sqrt{\gamma}$ to avoid diffusion effects which drive dissolution. Clearly, a setup that will fit all situations is difficult to pin down; our reference value = 50 also coincides with the membership of stellar associations seen in the low-density environments (Lada & Lada 2003; Gouliermis 2018). This value is also, in a loose sense, motivated by observations.

3.4. Dynamical mass estimates

We apply the MST algorithm to the subsample defined in Sect. 2.3 twice: once using the full (3D) spatial coordinates, and once using only projected coordinates on the xy plane. Three orthogonal views of the resulting groups identified in 3D are plotted in Fig. 3, and in 2D in Fig. 4. This leads to the identification of approximately 16 groups in each case. However, the comparison between these figures shows a different identification. In Fig. 3, the main over-densities of the region are well retrieved in all projection panels; we refer to these as *real groups*, as they represent the ideal identification with the MST algorithm. By contrast, in Fig. 4, group members appear well clumped only in the xy plane. On the other planes, they spread out over the whole area. These groups, henceforth *projected groups*, are biased in various ways: they can be *real* over-densities with some unrelated contaminants; or, completely artificial structures whose

³ There are a few exceptions, for instance: HDBSCAN cited above or the *friends-of-friends* algorithm (Huchra & Geller 1982) which both use spatial- and kinematic coordinates, but they are difficult to set up.

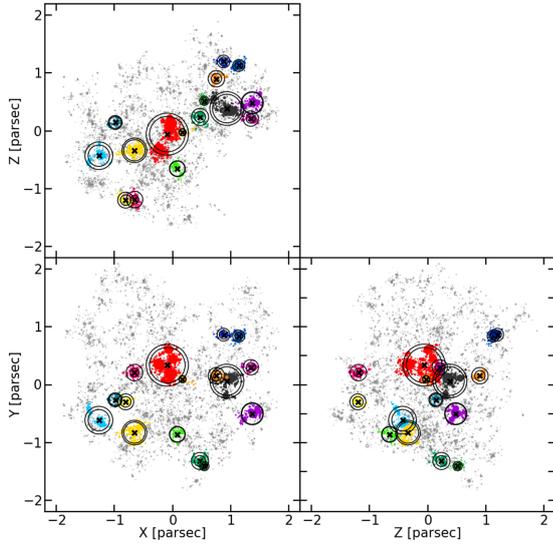


Fig. 3. Three orthogonal views of the sub-sample with real groups identified by the MST in three dimensions. The black crosses show the location of the centre of mass of each group, and the circles show their 75% and 90% Lagrangian radii.

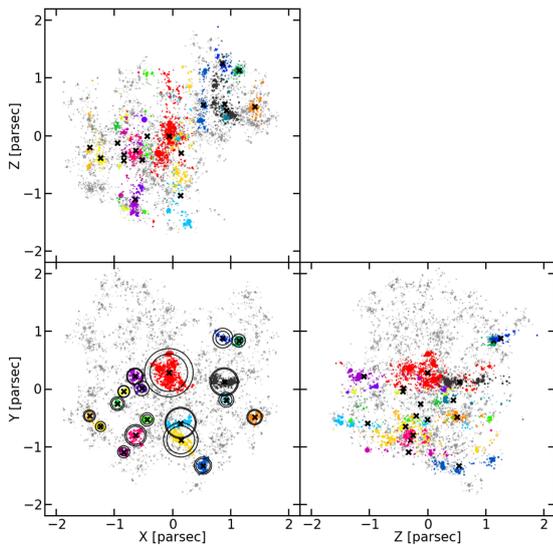


Fig. 4. Same as Fig. 3 but with projected groups identified by the MST in two dimensions along the xy plane (*lower left panel*). The projected Lagrangian radii are not tractable along the z -axis.

members are just aligned by chance in the line-of-sight. We emphasize the fact that this is different from the classical contamination by field stars, because all stars in our model are lying within the same star forming region of ≈ 4 parsec in diameter. Real and projected groups cannot be compared one to one. In order to compare both sets statistically, we quantify an observable property of these groups self-consistently.

With this in mind, we compute the dynamical mass which is a powerful tool to gauge the equilibrium state of a system (Portegies Zwart et al. 2010). It is also used to evaluate the dynamical state of young stellar systems and determine their fate (see e.g., Kuhn et al. 2019). From the virial theorem, we define the dynamical mass M_{dyn} :

$$M_{\text{dyn}} = \eta \times \frac{R_{\text{eff}} \sigma_{\text{LOS}}^2}{G}, \quad (4)$$

where R_{eff} is the effective radius, and σ_{LOS} is the line-of-sight velocity dispersion. The mass-to-light ratio, denoted η , depends on the phase-space configuration of the stellar system (Boily et al. 2005; Fleck et al. 2006). Hence all quantities that appear on the right-hand side of Eq. (4) are observables from projected data. This is crucial to compare M_{dyn} for real and projected groups in a self-consistent way.

As groups in our sample are asymmetric and not fully relaxed, we first determine the value of η empirically on real groups. This will constitute a value of reference to assess the difference between real and projected groups (we discuss the value of η further in Sect. 5.4).

To obtain an ensemble of η s representative of each group, we estimate the dynamical mass of real groups from a large number of different lines of sight, representing the possible viewing angles all over the sphere. To pick isotropic lines of sight, we chose the *HEALPix* tessellation (Górski et al. 2005) with the resolution parameter $N_{\text{side}} = 3$, giving 192 different angles of view separated by $\approx 14^\circ$ each. Our procedure iterates over these lines of sight and re-computes the position and velocity of the whole sample to estimate the projected dynamical mass of real groups corresponding to each view. Finally, since the *HEALPix* tessellation is symmetric with respect to the centre of the sphere, each configuration is paired with its mirror image. As a result there are $192/2 = 96$ independent mass estimates for each real group.

The effective radius in Eq. (4) is often replaced with the half-light radius in observational surveys of star clusters. The half-light radius matches the half-mass radius of analytic King-Michie cluster models in equilibrium (Mengel et al. 2002; Wolf et al. 2010) which allows direct comparison between observations and theory. However, this no longer holds true when the stellar clusters are mass-segregated (Gaburov & Gieles 2008). Moreover, the MST identifies asymmetric groups for which only one radius may not be representative of its size. Therefore, we explored different definitions of group radius corresponding to the projected Lagrangian radii at 50%, 75%, 90%, and 100%.

Computing the line-of-sight velocity dispersion of groups identified by the MST is not trivial. Indeed groups are in close interaction with their environment and members nearer to the edge are likely to be affected by the surroundings. We address this issue by estimating four different velocity dispersions corresponding to the stars lying within the four Lagrangian radii computed above. We also chose to compute the velocity dispersion using the biweight scale estimator (Beers et al. 1990) which is more robust than the standard deviation when the distribution contains outliers.

4. Assessing projection bias

4.1. Real groups

We thus obtain four different sets containing $16 \times 96 = 1536$ different estimates of the dynamical masses of real groups. These are plotted in Fig. 5. The distributions of all these estimates for each group are summarised as box-plots, where the median is shown as an orange vertical tick mark; the 16th and 84th percentiles by the box edges; and the extrema by the whiskers. The slope of the correlation between the dynamical and real mass in each panel matches the corresponding value of η for each radius. We compute this slope by a least-squares linear regression fit in each case. We see a clear correlation present at all radii, with a constant relative uncertainty of $\delta\eta/\eta \approx 0.12$, an indication that the correlation still holds even when including stars at the group edges.

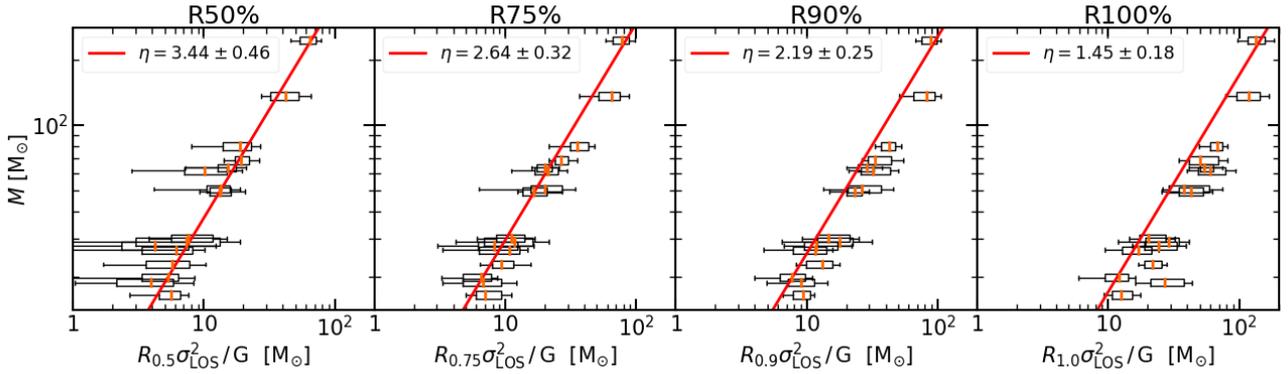


Fig. 5. Correlation between the true and dynamical mass of real groups. Each panel shows different chosen radii for the dynamical mass estimate. All possible values along different projecting angles for the same group are shown in a box-plot: the orange vertical tick marks represents the median of the distribution; the edge of the box, the 16th and 84th percentiles (such that 68% of the data are inside the box), and the horizontal bars correspond to the total width of the full distribution. The red lines are the least square fits on the data giving the values of η corresponding to each radius.

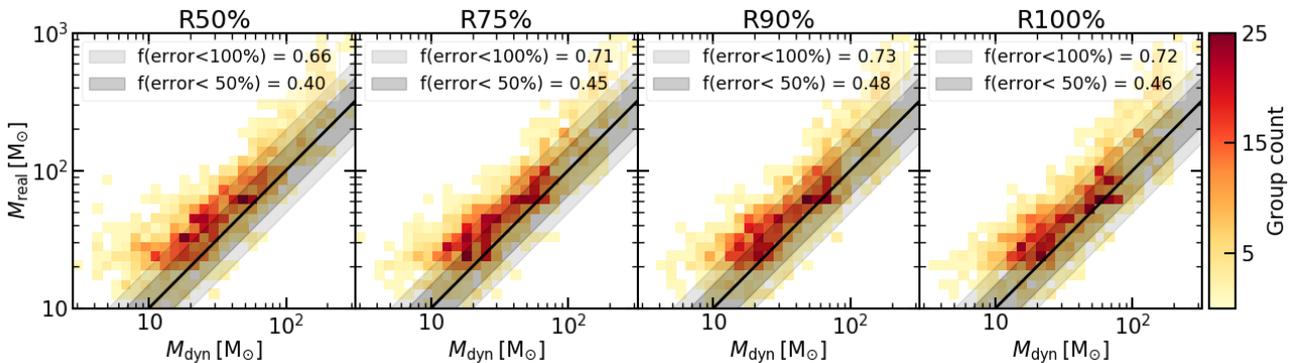


Fig. 6. Correlation between the true and dynamical masses of groups identified in projection represented as a 2D histogram. The black line shows the expectation constrained from real 3D groups (see Fig. 5). Different shaded areas show the fraction of well-estimated group masses within an error of 50% and 100%.

4.2. Projected groups

We focus now on the projected groups. We re-use the 96 viewing angles of the whole region computed in Sect. 3.4 to reach a similar number of mass estimates. As changing the line-of-sight will result in a different identification of projected groups, we re-compute the MST in each projection. Small variations in the number of groups identified in each projection lead to a different total number of projected groups of 1425. Their dynamical masses were estimated with the corresponding η factors constrained above (Sect. 3.4). The correlation with their real masses is plotted in Fig. 6 as a two-dimensional histogram. The black line shows the correlation derived from real groups. However, all mass estimates are shifted to lower values, falling to the left of the one-to-one relation on the figure. This highlights the fact that the mass of a group is underestimated when viewed in projection, irrespective of its true mass. The trend is not different from one panel to another, confirming that it is not sensitive to the choice of the radius. Globally, the deficit is greater than 50% for more than half of the groups, reaching 100% for a quarter of them. We attribute this systematic error to the misidentification of clumps due to projection effects. As all radii are affected, the centre and the outskirts of projected groups are similarly impacted.

To further investigate the reason for this deviation, we also compared radii and velocity dispersions of real and projected groups with similar masses. We divided both sets into three bins

in terms of mass: $[15; 35] M_{\odot}$, $[35; 100] M_{\odot}$ and $[100; 300] M_{\odot}$. The results corresponding to the 90% Lagrangian radii are plotted in Fig. 7, but other radii give similar trends. We take a statistical approach to explore differences between real and projected groups. The distributions within each mass bin are represented by box plots of different colours: black for real groups and blue for projected groups. We recover on the left panel the systematic error in dynamical mass between real and projected groups (previously seen on Fig. 6). The two other panels show the respective distributions of radii and velocity dispersions used to compute the dynamical mass. The properties of projected groups in a given mass bin are statistically lower than those of real groups. Moreover, the trend is more pronounced for the most massive groups. For the less massive groups, their Lagrangian radii are globally comparable whereas their velocity dispersions are systematically about 20% lower. We note here that this is not the case when choosing only the stars lying in the 50% or 75% Lagrangian radii where Poissonian fluctuations can increase the difference up to $\approx 30\%$. For groups more massive than $100 M_{\odot}$, both radii and velocity dispersions are equally shifted by about 30%, independently of the chosen radius.

In summary, projection effects lead to significant underestimation of both the size and velocity dispersion of projected groups. Only the radii of the less massive groups are nearly not affected. Some may find it surprising that the velocity dispersion is lowered in projected groups where some stars are not part of any clumps. However, we must bear in mind that these

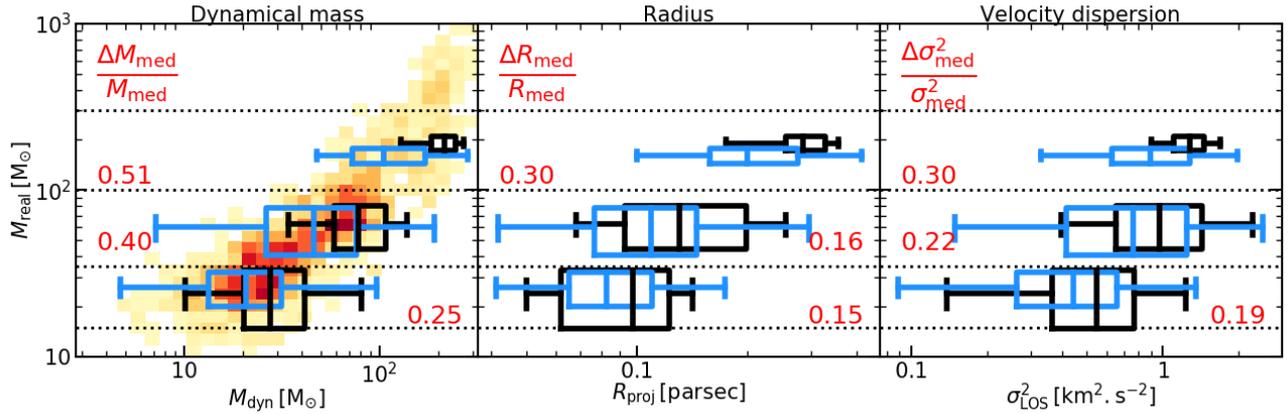


Fig. 7. Comparison between projected and real group properties for three mass bins: $[15; 35] M_{\odot}$, $[35; 100] M_{\odot}$ and $[100; 300] M_{\odot}$ (dashed lines). Distributions of each estimate are represented as black box plots for real groups and blue ones for projected groups. The relative error on the median for each distribution is annotated in red. In each box plot, the vertical middle tick represents the median of the distribution, the edges show the 16th and 84th percentiles, and the whiskers correspond to the full extension of the distribution. The estimates plotted correspond to the 90% Lagrangian radii but other radii give similar trends. The height of the boxes is proportional to the number of groups inside each mass bin. The 2D histogram of Fig. 6 is over-plotted on the panel of the dynamical mass estimate.

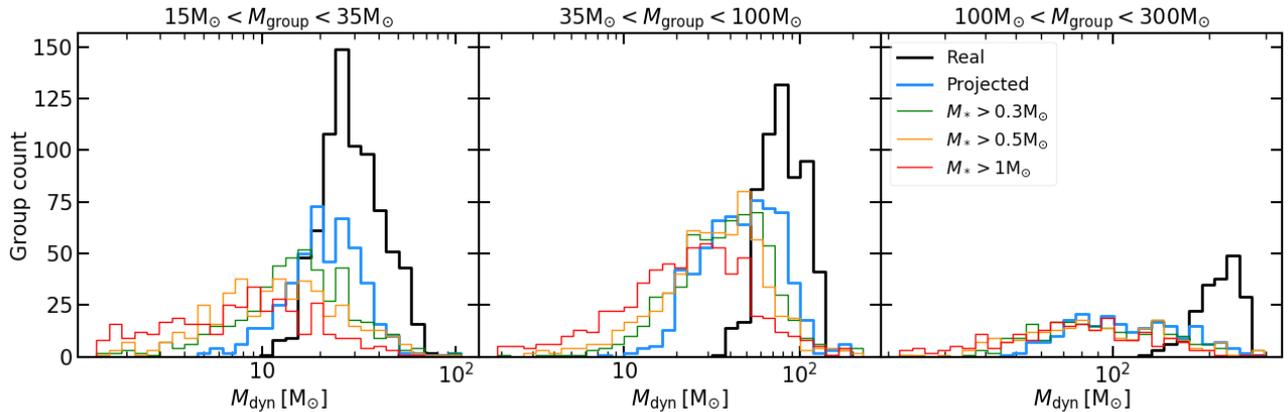


Fig. 8. Comparison of the dynamical mass estimated on projected groups with different levels of incompleteness. The three panels correspond to the different real mass bins chosen in Fig. 7. The black histogram is the reference set of real groups and the blue histogram is the set of complete projected groups. The histograms in green, orange and red represents projected groups that are only complete down to $0.3 M_{\odot}$, $0.5 M_{\odot}$ and $1 M_{\odot}$ respectively. The figure shows the results using the 90% Lagrangian radii.

stars are in the same region, and therefore they are likely to have a velocity close to the mean value (of $\approx 0 \text{ km s}^{-1}$ in our case, see Sect. 5.1), which is the reason for the lower velocity dispersion in projected groups.

4.3. Incompleteness

Projection effects are always present in astronomical observations where spherical symmetry is a valid ansatz. This eases comparison with theoretical models. However, this assumption is not justified for young systems. We indeed show that a lack of precise information in the line of sight can introduce systematic biases in the clump properties.

Another common bias when studying young stellar systems arises from incompleteness of the target populations. This also affects particularly star forming regions where extinction can be significant. We may estimate the importance of the completeness of the sample in comparison with that of the morphological projection biases that we have been discussing so far. At first order, the faintest sources are also the least massive ones, and so we apply different mass cuts to the set of projected groups (at $0.3 M_{\odot}$, $0.5 M_{\odot}$ and $1 M_{\odot}$). Again, we estimated the properties

of the resulting groups on the different lines of sight. Figure 8 shows the resulting dynamical mass distribution within the three mass bins defined earlier in Fig. 7. Distributions of real groups and complete projected groups from Fig. 7 are also plotted for comparison. We observe that the distribution of dynamical mass estimates is progressively shifted to smaller values when incompleteness increases. This trend is expected, because the more massive stars selected are segregated in space over a smaller volume (Dorval et al. 2016), while their velocity dispersion will also be lower due to star–star interactions. As incompleteness of the sample will always increase the scatter owing to lower number statistics, dynamical mass distributions of incomplete groups show greater dispersion.

This analysis is just the first step in a longer process, as we know that the problem of incompleteness is complex. Detection limits of observations is wavelength dependant and extinction is never constant inside the same star forming regions. Indeed, dust clouds also harbour filamentary structures in interaction with newly born stars and dramatically modify the aspects of these clusters. Other important additions are the stellar photometry and interstellar reddening, two quantities affecting the sampling of stars with opposite results. Indeed, the most massive

stars are the brightest but are also likely to be the most embedded (Motte et al. 2018). A self-consistent simulation including hydrodynamics would be needed to more accurately explore these aspects.

5. Discussion

5.1. Decontamination

We recall that the stars contaminating the set of projected groups are stars from the same region, separated only by a few parsecs and sharing the same global dynamical state. Therefore, it can be difficult to distinguish between contaminants and group members based on their kinematics alone. We ask if *Gaia* DR2 proper motions can separate two groups seen in projection using their individual proper motions. If we use a reference distance to Orion of 400 pc (Kounkel et al. 2018), the highest accuracy of *Gaia* DR2 proper motions is 0.1 km s^{-1} at $M_G = 13$ but only reaches 1 km s^{-1} at $M_G = 18$. As *Gaia* EDR3 will improve this value by a factor of two, we can expect to distinguish between two groups separated by a systematic motion of more than 0.5 km s^{-1} .

Figure 9 shows a one-dimensional position–velocity plot of stars in real (3D) groups. Each point is colour-coded as in Fig. 3. Field stars are shown in light grey. The filled circles are the mean values for each group and the error-bars represent the standard deviation, which is an estimate of the typical internal motion of the clumps. An indication of the relative motions between these groups is given by the standard deviation of these mean values, which is $\approx 0.4 \text{ km s}^{-1}$. This is of the same order as the internal motion, which is $\approx 0.5 \text{ km s}^{-1}$. We conclude that the majority of real groups cannot be differentiated. We find only one instance where two out of sixteen groups are separated by $\approx 2 \text{ km s}^{-1}$ which is within the *Gaia* detection limit. This is by construction a feature brought by our GDF method, because we did not introduce large-scale motion in our fragmentation method. Such large-scale modes can be included in a future study.

5.2. Varying the total number of stars in the region

Although the reference set of 15 000 stars we previously studied allows us to explore clumps of different membership and density, the large projected density of $\approx 500 \text{ stars pc}^{-2}$ can have an impact on the level of contamination of these clumps. Moreover, such clumpy configurations are not always seen in star forming regions. However, it is clear that none of the star forming regions are perfectly isolated (Kuhn et al. 2014). The local environment of a particular region, even if it is not as structured as in our reference model, can still introduce some biases when inferring its properties. To probe such conditions, we built a GDF model with only 1000 stars and repeated the analysis. The sub-selection method with the HOP algorithm (Sect. 2.3) extracts a sample of 765 stars for a surface density of 50 stars pc^{-2} . This new model is, as expected, less structured: only one major overdensity developed at the centre of the region (see upper panel of Fig. 10). We therefore focus on this single clump detected by the MST. However, some structures exist in its environment that may contaminate it when seen in projection. We performed the estimation of the dynamical mass of this single group as in Sect. 3.4. We used the values of η constrained previously in Sect. 3.4. We also identify groups in projection for the same 96 lines of sight⁴

⁴ We note here that the MST identifies sometimes more than one group in projection. In such cases, we only retain the most massive one.

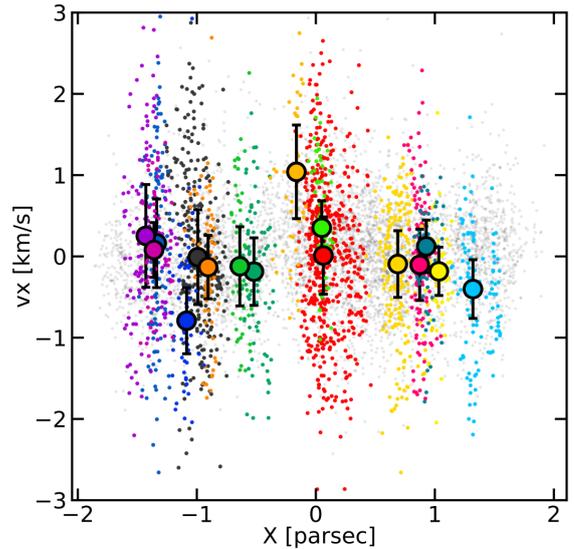


Fig. 9. Position–velocity plot of stars in the sample. Stars that do not belong to any group are coloured in light grey whereas those in real 3D groups match the colours used in Fig. 3. The filled circles are the mean values for each group and the error-bars represent the standard deviation.

and estimate their dynamical mass too. Finally we compare the distribution of both estimations. The distributions corresponding to the 75% Lagrangian radii can be seen in the bottom panel of Fig. 10. Other radii show similar behaviours. The true mass of the group is marked as the red line for reference. The results are similar to those shown in Fig. 8: the dynamical mass of the projected group is systematically lower than that of the real group. Therefore, even the smaller number of sub-structures in this model can still lead to inaccurate estimation of clump properties. Second, although the value of η was not calibrated on this model, the dynamical mass is well retrieved in 3D. Our overall conclusions are therefore not sensitive to the precise surface density at least from 50 stars pc^{-2} to $500 \text{ stars pc}^{-2}$.

5.3. Clump-finding algorithm

Our study specifically aims to assess the bias introduced by projection effects within a star forming region. We show that projection effects systematically lower the spatial and kinematic properties of clumps. However, the amplitude of this shift is difficult to quantify. The variation that we estimate to range from 10 to 30% is due to the intrinsic nature of projection bias, which depends on the shape and orientation of a particular region. It is also linked to the method we choose.

However, most clump-finding algorithms also use projected spatial coordinates. For instance Joncour et al. (2018) proposed a method to calibrate the DBSCAN algorithm using a statistical approach. The DBSCAN is a close alternative to the MST. These latter authors exploit the nearest-neighbour statistics of a 2D Poissonian distribution to compute the parameters of the DBSCAN. This allows them to select only projected overdensities at three-sigma level over random expectation. Nevertheless, this method is most suitable for small-scale structures of approximately ten members, where contamination in the line of sight is weak. For bigger clumps, like those that we identified in the present study, projection effects are more significant. Therefore, we may expect that such a method will lead to similar results to the MST approach adopted here.

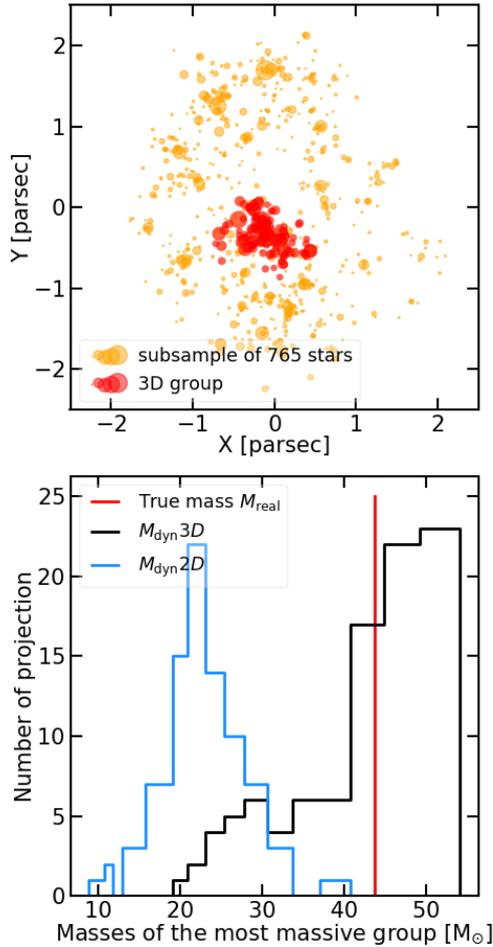


Fig. 10. *Upper panel:* sub-sample extracted from a GDF model of 1000 stars with the method of Sect. 2.3. Only one major group is identified in 3D by the MST (red dots). *Lower panel:* distributions of the dynamical mass estimated on different projections of the main group identified by the MST in 3D (black line) and in projection (blue line). The true mass is marked as the vertical red line.

A step forward would be to build a set of GDF models reproducing configurations of a wide range in density, and train a machine-learning algorithm on this sample to bridge the 3D identifications to 2D projections with a significance estimator, similarly to what was developed by Joncour et al. (2018) for DBSCAN. We show here that all groups show comparable projection biases, with a trend of increasing amplitude with group mass. This suggests that a data set of models with reasonably small groups –of up to 1000 stars– should still yield useful statistics while remaining manageable in terms of volume. Further work in that direction is underway.

5.4. The mass-to-light ratio η

A secondary result emerging from this study is the value of the mass-to-light ratio η linking the true and dynamical masses of real groups (see Sect. 3.4). The value $\eta \sim 10$ is often chosen to match a wide range of spherical King-Mitchie models (Spitzer 1987; Mengel et al. 2002). However, Fleck et al. (2006) already showed that η varies tremendously through the lifetime of a cluster, mostly as a result of mass segregation.

The value $\eta \simeq 10$ is about a factor three greater than our evaluation of ≈ 3.4 for real groups. Previous works (see e.g.,

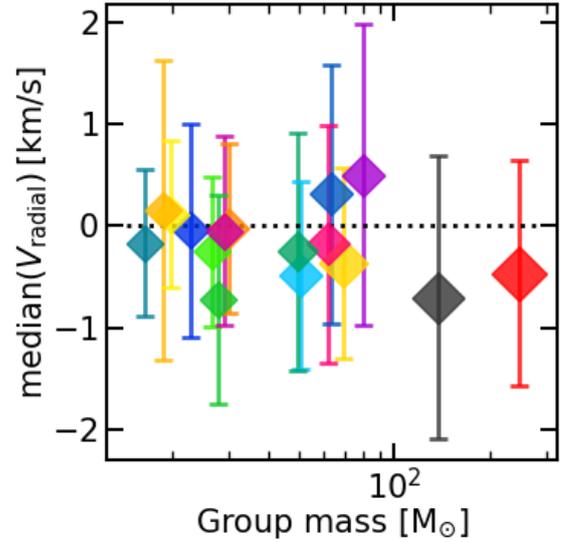


Fig. 11. Median of the radial velocity of stars in real groups with respect to their centre of mass. The colours are chosen to match those in Fig. 3.

Portegies Zwart et al. 2010; Parker & Wright 2016) predicted that low values of η could be a signature property of young systems, one related to their state of equilibrium. As nothing suggests that real groups are in virial equilibrium, such a low value of η could be an indication of expansion. We rule out this explanation by computing the radial velocities of stars in real groups from their respective centres of mass. The median for each group is plotted in Fig. 11. It shows that real groups are in a quasi-steady state, with $\bar{v}_r \approx 0 \text{ km s}^{-1}$. This is perhaps surprising, because they are in close interaction with their environment, for instance through the tidal field of neighbouring groups and fly-by stars. However, it must be borne in mind that these configurations are a snapshot taken after only $t \simeq 0.5 \text{ Myr}$. The age of the system is of the same order as the dynamical crossing time for groups, when little internal kinetic energy diffusion has had time to drive expansion.

Precise determination of η should help observers to put further observational constraints on the dynamical state of sub-clusters in star forming regions. For instance Kuhn et al. (2019) recently computed the *virial-expected* velocity dispersion of 19 young clusters. A lack of constraints on the value of η has prevented confirmation of the equilibrium dynamical state of these systems. Assuming a value of 10 lead Kuhn et al. (2019) to conclude that all of these clusters were unbound, although their *Gaia* proper motions did not show clear signs of expansion. Using a reduced value of 5, closer to our evaluation, would reconcile these contradictory results. We stress that the systems that we have analysed, which are of linear scale $\approx 0.5 \text{ pc}$ or less, are small compared to those of Kuhn et al. (2019). To make a more meaningful comparison with their survey would require that we develop models of a full-size $\approx 100 \text{ pc}$ region, with disc-shearing and the vertical disc structure of the Milky Way added to the GDF modelling of Sect. 2. We hope to report on these aspects in a future contribution.

6. Conclusions

In the present study, we use GDF models to self-consistently reproduce a fragmented distribution of stars to assess the systematic bias introduced by an imprecise identification of stellar clumps in projection compared to an ideal one in 3D. To this end,

we performed a statistical comparison between the properties of stellar over-densities (of membership $N_{\text{stars}} > 50$) identified in 3D and in projection by a generic MST algorithm within a single star forming region of ≈ 4 pc in width⁵. Our main results can be summarised as follows:

1. We highlight that the dynamical mass of clumps identified in projection is systematically underestimated compared to the ideal 3D identification. This systematic error is greater than 50% in about half of the cases and can reach up to 100% in a quarter of them. This is due to the incorrect selection of the clump-finding algorithm when dealing only with projected coordinates in such fragmented regions.
2. This projection bias impacts both radius and velocity dispersion estimates of projected groups. Eliminating potential outliers at the edges of the groups identified by the MST did not lead to any improvements.
3. We notice a trend ranging from the properties of less massive ($M < 100 M_{\odot}$) groups, which are less affected or even not at all, to those of the most massive groups ($M > 100 M_{\odot}$), which are more systematically affected. For these latter, radii and velocity dispersions are statistically 30% lower.
4. As the projection effect is not the only bias in star forming regions, we also considered simple prescriptions to account for incompleteness in the observations. We show that sample incompleteness increases the systematic error. However, its impact is of a different nature and can be seen as a blurring effect. For example, the most massive groups are almost unaffected since they still contain a sufficient number of massive members for them to be characterised.
5. We conclude that projection effects are the main limitation in estimating properties of clumps identified within a single fragmented region with no or insufficiently accurate distance estimates. For most nearby star forming regions, the best *Gaia* parallaxes have uncertainties of a few percent. For these distant regions, that is, at more than 100 pc, this prevents any clump-finding algorithm from properly defining clumps of less than a few parsecs in size. Accurate identification of such clumps is a key challenge in better constraining the processes of star formation. This also has major implications for predicting the future of these young systems in bound clusters or unbound associations.
6. We also show that GDF models offer an efficient method to initialise N -body simulations of young clusters. With this method, we are able to explore the structures built by gravity alone in more detail in order to better understand the role of gravity compared to the many processes at play in star forming regions.

Acknowledgements. We thank the referee for the constructive remarks which led to improve the quality of the paper. Timothé Roland also acknowledges Paolo Bianchini, Caroline Bot, Joe Lewis and Jonathan Chardin for their helpful comments and suggestions. This work has used the AMUSE framework (<https://github.com/amusecode/amuse>) (Portegies Zwart & McMillan 2018; Portegies Zwart et al. 2009, 2013; Pelupessy et al. 2013) and the Python programming language with the following packages: `matplotlib` (Hunter 2007), `NumPy` (Harris et al. 2020), `SciPy` (Virtanen et al. 2020) and `Astropy` (Astropy Collaboration 2013, 2018).

References

Allison, R. J., Goodwin, S. P., Parker, R. J., et al. 2009, *MNRAS*, **395**, 1449
 Alves, J., Zucker, C., Goodman, A. A., et al. 2020, *Nature*, **578**, 237

⁵ We recall that we did not take into account any additional foreground or background contamination besides the one introduced by the region itself.

- André, P., Men'shchikov, A., Bontemps, S., et al. 2010, *A&A*, **518**, L102
 André, P., Arzoumanian, D., Könyves, V., Shimajiri, Y., & Palmeirim, P. 2019, *A&A*, **629**, L4
 Andre, P., Ward-Thompson, D., & Barsony, M. 2000, in *Protostars and Planets IV*, eds. V. Mannings, A. P. Boss, & S. S. Russell, 59
 Astropy Collaboration (Robitaille, T. P., et al.) 2013, *A&A*, **558**, A33
 Astropy Collaboration (Price-Whelan, A. M., et al.) 2018, *AJ*, **156**, 123
 Bastian, N., Covey, K. R., & Meyer, M. R. 2010, *ARA&A*, **48**, 339
 Bate, M. R. 2012, *MNRAS*, **419**, 3115
 Beers, T. C., Flynn, K., & Gebhardt, K. 1990, *AJ*, **100**, 32
 Beuret, M., Billot, N., Cambrésy, L., et al. 2017, *A&A*, **597**, A114
 Boily, C. M., Lançon, A., Deiters, S., & Heggie, D. C. 2005, *ApJ*, **620**, L27
 Cantat-Gaudin, T., Jordi, C., Wright, N. J., et al. 2019, *A&A*, **626**, A17
 Cartwright, A., & Whitworth, A. P. 2004, *MNRAS*, **348**, 589
 Chabrier, G. 2003, *PASP*, **115**, 763
 Dorval, J., Boily, C. M., Moraux, E., Maschberger, T., & Becker, C. 2016, *MNRAS*, **459**, 1213
 Dorval, J., Boily, C. M., Moraux, E., & Roos, O. 2017, *MNRAS*, **465**, 2198
 Eisenstein, D. J., & Hut, P. 1998, *ApJ*, **498**, 137
 Fariás, J. P., Tan, J. C., & Chatterjee, S. 2019, *MNRAS*, **483**, 4999
 Fleck, J. J., Boily, C. M., Lançon, A., & Deiters, S. 2006, *MNRAS*, **369**, 1392
 Foster, J. B., Cottaar, M., Covey, K. R., et al. 2015, *ApJ*, **799**, 136
 Fujii, M. S., & Portegies Zwart, S. 2016, *ApJ*, **817**, 4
 Gaburov, E., & Gieles, M. 2008, *MNRAS*, **391**, 190
 Goodwin, S. P., & Whitworth, A. P. 2004, *A&A*, **413**, 929
 Górski, K. M., Hivon, E., Banday, A. J., et al. 2005, *ApJ*, **622**, 759
 Gouliermis, D. A. 2018, *PASP*, **130**, 072001
 Großschedl, J. E., Alves, J., Meingast, S., et al. 2018, *A&A*, **619**, A106
 Grudić, M. Y., Hopkins, P. F., Faucher-Giguère, C.-A., et al. 2018, *MNRAS*, **475**, 3511
 Gutermuth, R. A., Megeath, S. T., Myers, P. C., et al. 2009, *ApJS*, **184**, 18
 Hacar, A., Tafalla, M., & Alves, J. 2017, *A&A*, **606**, A123
 Hacar, A., Tafalla, M., Forbrich, J., et al. 2018, *A&A*, **610**, A77
 Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, *Nature*, **585**, 357
 Hennebelle, P., Commerçon, B., Chabrier, G., & Marchand, P. 2016, *ApJ*, **830**, L8
 Huchra, J. P., & Geller, M. J. 1982, *ApJ*, **257**, 423
 Hunter, J. D. 2007, *Comput. Sci. Eng.*, **9**, 90
 Hut, P., Makino, J., & McMillan, S. 1995, *ApJ*, **443**, L93
 Joncour, I., Duchêne, G., Moraux, E., & Motte, F. 2018, *A&A*, **620**, A27
 Kirk, H., & Myers, P. C. 2011, *ApJ*, **727**, 64
 Kounkel, M., & Covey, K. 2019, *AJ*, **158**, 122
 Kounkel, M., Covey, K., Suárez, G., et al. 2018, *AJ*, **156**, 84
 Kroupa, P., Aarseth, S., & Hurley, J. 2001, *MNRAS*, **321**, 699
 Krumholz, M. R. 2014, *Phys. Rep.*, **539**, 49
 Kuhn, M. A., Feigelson, E. D., Getman, K. V., et al. 2014, *ApJ*, **787**, 107
 Kuhn, M. A., Hillenbrand, L. A., Sills, A., Feigelson, E. D., & Getman, K. V. 2019, *ApJ*, **870**, 32
 Lada, C. J., & Lada, E. A. 2003, *ARA&A*, **41**, 57
 Maschberger, T. 2013, *MNRAS*, **429**, 1725
 Maschberger, T., & Clarke, C. J. 2011, *MNRAS*, **416**, 541
 Maschberger, T., Clarke, C. J., Bonnell, I. A., & Kroupa, P. 2010, *MNRAS*, **404**, 1061
 McMillan, S. L. W., & Hut, P. 1996, *ApJ*, **467**, 348
 Mengel, S., Lehnert, M. D., Thatte, N., & Genzel, R. 2002, *A&A*, **383**, 137
 Meylan, G., & Heggie, D. C. 1997, *A&ARv*, **8**, 1
 Moeckel, N., & Bate, M. R. 2010, *MNRAS*, **404**, 721
 Montillaud, J., Juvela, M., Vastel, C., et al. 2019, *A&A*, **631**, A3
 Motte, F., Bontemps, S., & Louvet, F. 2018, *ARA&A*, **56**, 41
 Parker, R. J., & Wright, N. J. 2016, *MNRAS*, **457**, 3430
 Pavlík, V., Kroupa, P., & Šubr, L. 2019, *A&A*, **626**, A79
 Pelupessy, F. I., van Elteren, A., de Vries, N., et al. 2013, *A&A*, **557**, A84
 Pokhrel, R., Myers, P. C., Dunham, M. M., et al. 2018, *ApJ*, **853**, 5
 Portegies Zwart, S., & McMillan, S. 2018, *Astrophysical Recipes: The Art of AMUSE*, 2514-3433 (AAS IOP Astronomy Publishing)
 Portegies Zwart, S., McMillan, S., Harfst, S., et al. 2009, *New Astron.*, **14**, 369
 Portegies Zwart, S. F., McMillan, S. L. W., & Gieles, M. 2010, *ARA&A*, **48**, 431
 Portegies Zwart, S., McMillan, S. L. W., van Elteren, E., Pelupessy, I., & de Vries, N. 2013, *Comput. Phys. Commun.*, **184**, 456
 Spitzer, L. 1987, *Dynamical Evolution of Globular Clusters* (Princeton: Princeton University Press)
 Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, *Nat. Meth.*, **17**, 261
 Wall, J. E., McMillan, S. L. W., Mac Low, M.-M., Klessen, R. S., & Portegies Zwart, S. 2019, *ApJ*, **887**, 62
 Wall, J. E., Mac Low, M. M., McMillan, S. L. W., et al. 2020, *ApJ*, accepted [arXiv:2003.09011]
 Wolf, J., Martinez, G. D., Bullock, J. S., et al. 2010, *MNRAS*, **406**, 1220

PROPRIÉTÉS DYNAMIQUES DES GROUPES

PRODUITS PAR LES MODÈLES GDF

Dans le chapitre 4 précédent, nous avons pointé une limitation à l'identification de sous-structures stellaires dans les régions de formation d'étoiles (voir aussi [Zamora-Avilés et al., 2017](#)). Néanmoins l'amélioration des techniques de détection couplée à la réduction des incertitudes de mesures conduira dans un futur proche, à limiter ce problème. Ainsi la mesure de l'état dynamique de ces petits ensembles permettrait de contraindre le lien manquant entre la dynamique du gaz en effondrement et celle des étoiles nouvellement créées. Bien que reposant sur des hypothèses, les simulations restent de première importance dans le but de mieux contraindre les propriétés dynamiques de ces petits groupes. Dans le cas des modèles GDF ce lien entre la structure fragmentée de l'amas et les surdensités stellaires qui y émergent est entièrement dû à la gravité. Ainsi l'étude de ces petits groupes permet d'isoler les propriétés seulement fonction de la dynamique N -corps.

Nous explorons dans ce chapitre les propriétés dynamiques des groupes identifiés en trois dimensions au sein du modèle présenté dans l'article, au chapitre précédent. Nous ferons notamment le lien entre leurs propriétés intrinsèques et celles qui pourraient être extraites par des observations. Puis nous les comparerons à un ensemble d'amas issus du catalogue DR2 de la mission *Gaia*.

5.1 Rapport masse/lumière η

Nous nous intéresserons donc ici aux groupes identifiés en trois dimensions par le MST au chapitre 4 (voir Fig. 5.1). Ces derniers montrent une bonne corrélation entre masse réelle et masse dynamique (voir la Fig.5 de l'article). Cette relation linéaire possède un facteur de proportionnalité communément appelé rapport masse/lumière (η) en ce qu'il permet de contraindre la masse d'un système basé sur des observations. Cette corrélation est déduite du théorème du viriel. Considérant ainsi un système sphérique et à l'équilibre :

$$\begin{aligned} 2E_k + E_p &= 0 \\ M\sigma^2 - \frac{GM^2}{r_g} &= 0 \end{aligned} \quad (5.1)$$

où σ^2 est une mesure de la dispersion des vitesses 3D au sein de l'amas et r_g le rayon gravitationnel. La forme théorique de la masse dynamique s'exprime :

$$M_{\text{dyn}} = \frac{r_g \sigma^2}{G} \quad (5.2)$$

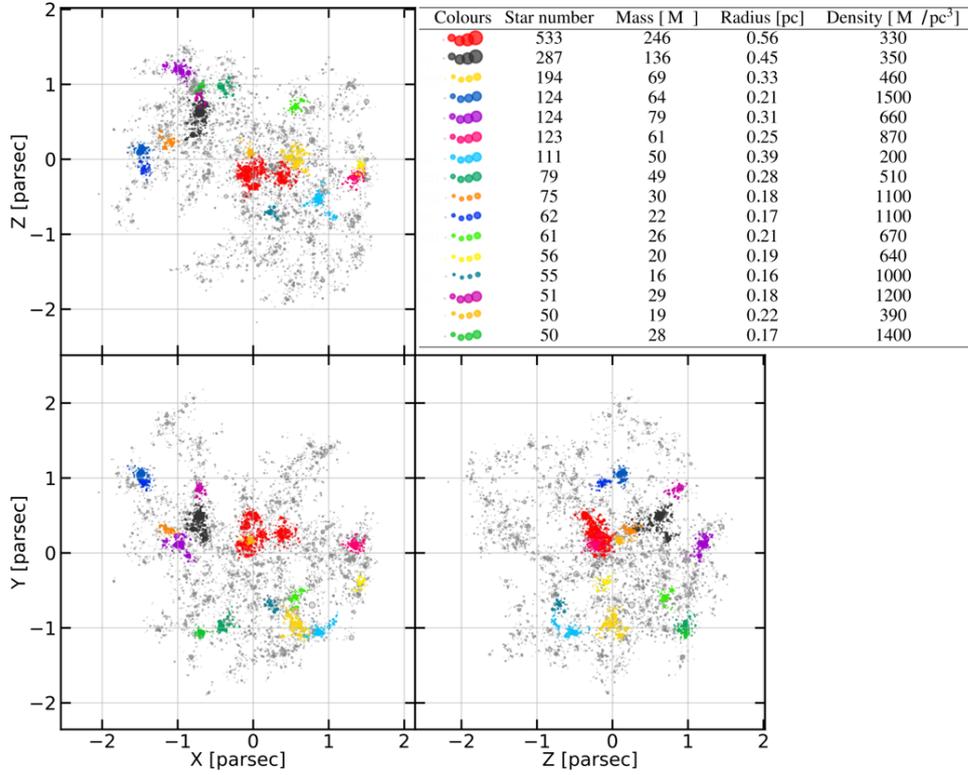


FIGURE 5.1 – Distribution des étoiles du modèle présenté au chapitre 4, vue selon trois projections orthogonales. La légende détaille quelques propriétés des groupes identifiés par le MST en 3D.

r_g représente un rayon théorique et dépend de la distribution de masse des étoiles au sein de l'amas. Pour des modèles de King ou de Plummer, sa valeur est bien contrainte et est reliée au rayon de mi-masse r_{hm} par un facteur ~ 2.5 (Portegies Zwart et al., 2010). Néanmoins r_{hm} et σ^2 sont des propriétés globales et non directement observables. À la manière du chapitre précédent, les observateurs mesurent plutôt le rayon projeté ou rayon cylindrique (que l'on symbolise par un grand R ou R_{2D}) et la dispersion des vitesses le long de la ligne de visée σ_{LOS}^2 .

Pour un grand nombre de modèles, Spitzer (1987) a contraint le rapport entre le rayon de mi-masse 3D r_{hm} et celui projeté R_{hm} à

$$\frac{r_{hm}}{R_{hm}} \approx 1.3. \quad (5.3)$$

D'autre part, la dispersion des vitesses (3D) est la moyenne de chacune de ses composantes selon les trois directions (ceci correspond plus à une définition d'usage). En supposant l'isotropie des vitesses on obtient :

$$\sigma^2 = 3 \times \sigma_{LOS}^2 \quad (5.4)$$

Ainsi en combinant ces facteurs, on obtient l'équation de la masse dynamique très généralement utilisée :

$$M_{dyn} = \eta \times \frac{R_{hm} \sigma_{LOS}^2}{G}, \quad (5.5)$$

avec ainsi le rapport masse/lumière,

$$\eta = \frac{r_g}{r_{hm}} \times \frac{r_{hm}}{R_{hm}} \times \frac{\sigma^2}{\sigma_{LOS}^2} \approx 2.5 \times 1.3 \times 3 \approx 10, \quad (5.6)$$

Alors que le premier facteur dépend de la distribution de masse au sein de l'amas considéré, les deux derniers ne sont dus qu'aux limites de mesure du rayon et de la dispersion des vitesses. Dans le but de mieux cerner les potentiels biais de mesure de ces propriétés, nous avons profité des nombreuses projections réalisées au chapitre 4 pour tester si ces différents facteurs pouvaient s'appliquer aux groupes identifiés en 3D sur les modèles GDF. La Fig. 5.2 montre la distribution des rayons estimés en projection sur chacun des groupes sous la forme de diagrammes « en violon ». Ces diagrammes montrent via la hauteur de la zone colorée, la densité de probabilité – échantillonnée dans notre cas sur 96 projections différentes – des valeurs de R_{2D} indiquées sur l'axe des abscisses. On observe qu'en moyenne, le facteur de conversion entre rayon 3D et 2D est légèrement inférieur à la valeur attendue de 1.3 pour le rayon à mi-masse. Pour les autres rayons englobant 75%, 90% et 100% de la masse, on obtient presque l'égalité entre les deux rayons. Néanmoins les distributions couvrent de larges plages de valeurs et sont hautement dissymétriques reflétant la structure complexe des groupes identifiés.

De la même manière, nous avons réalisé le même test concernant la dispersion des vitesses. La Fig. 5.3 montre la corrélation entre les deux mesures. La très bonne corrélation moyenne entre ces valeurs est ici le résultat du choix des lignes de visée. Ces dernières ont été adoptées dans le but de sonder la région de manière isotrope. Ainsi les différentes valeurs de σ_{LOS} mesurées représentent un bon échantillonnage de la dispersion des vitesses 3D de chaque groupe.

La bonne correspondance entre les valeurs attendues et contraintes sur les groupes de nos modèles concernant ces deux rapports indique que la valeur de ces deux facteurs n'est pas fortement sensible à la structure très asymétriques des groupes de nos modèles – du moins, en moyenne. En effet, la mesure de ces propriétés est en pratique, réalisée seulement sur une seule ligne de visée et les dispersions importantes que l'on observe Fig. 5.2 et Fig. 5.3 présage que cette mesure reste grandement imprécise.

La valeur du rapport masse/lumière contrainte au chapitre précédent ~ 3.44 est néanmoins largement inférieure à celle habituellement utilisé pour les études dynamiques d'amas (Portegies Zwart et al., 2010; Parker & Wright, 2016; Kuhn et al., 2019). Si les biais de projection ne semble pas jouer un effet important, d'autres hypothèses sont à étudier. En premier lieu, la distribution de masse au sein des groupes, directement liée au rapport entre le rayon gravitationnel et le rayon de mi-masse $\frac{r_g}{r_{hm}}$. Pour tester cette éventualité, nous avons calculé le rayon gravitationnel théorique à partir de l'énergie potentielle et la masse de chaque groupe (voir équation 5.1). La Fig. 5.4 montre la corrélation entre les deux rayons pour l'ensemble de ces groupes. On observe que le facteur de corrélation calculé sur nos groupes est en deçà de la valeur de 2.5 attendu pour des distributions de masse plus « classiques » (c.-à-d. de King ou de Plummer). Néanmoins, on observe que la situation est assez disparate selon les groupes identifiés, bien qu'issus du même modèle. La majorité des groupes qui sont aussi les moins massifs (mais aussi les plus denses), ont un rapport entre ces rayons plutôt en accord avec la valeur de 2.5. Ce qui n'est pas le cas des groupes plus massifs dont le facteur de corrélation est plus faible ~ 1.5 . Une inspection plus détaillée des groupes les moins massifs montre que leur rayon de mi-masse est ridiculement

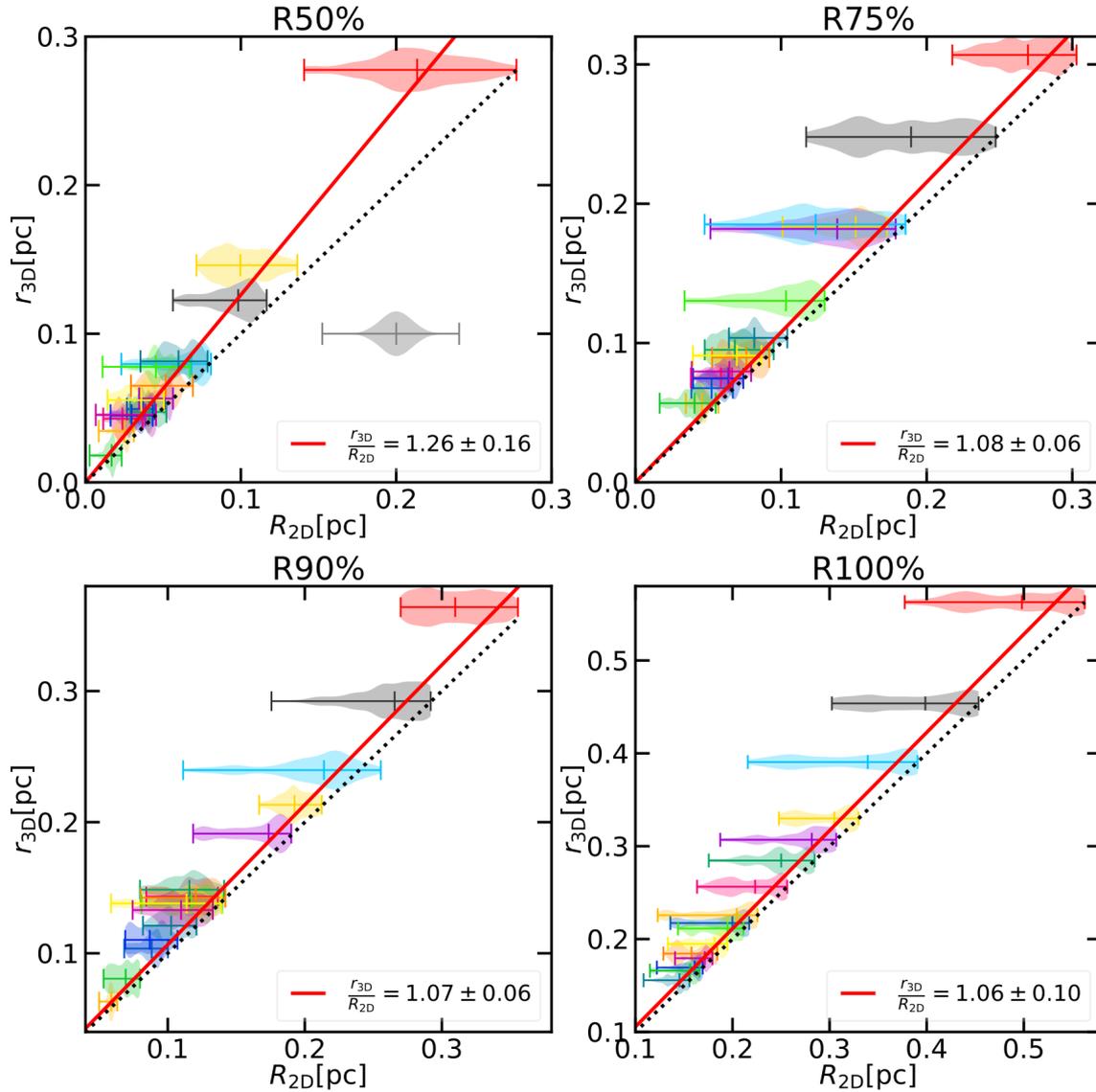


FIGURE 5.2 – Corrélation entre les rayons projetés et le rayon 3D des groupes identifiés en 3D par le MST au sein d'un amas fragmenté pour les rayons englobant 50%, 75%, 90% et 100% de la masse du groupe. La distribution des rayons estimés en projection pour chacun des groupes est représentée sous la forme d'un diagramme « en violon ». La ligne horizontale marque à l'endroit de la valeur du rayon 3D r_{3D} , l'extension totale de la distribution. Les marques verticales montrent respectivement, les valeurs minimales, médianes et maximales de R_{2D} . Pour comparaison, une distribution gaussienne centrée en 0.2 d'une largeur à mi-hauteur égale à 0.01 est représentée en gris clair sur le panneau en haut à gauche. Les couleurs choisies pour chaque groupe correspondent à celles utilisées Fig. 5.1. La droite rouge montre la régression linéaire réalisée entre rayons 3D et rayons projetés tandis que la droite noire pointillée marque l'égalité.

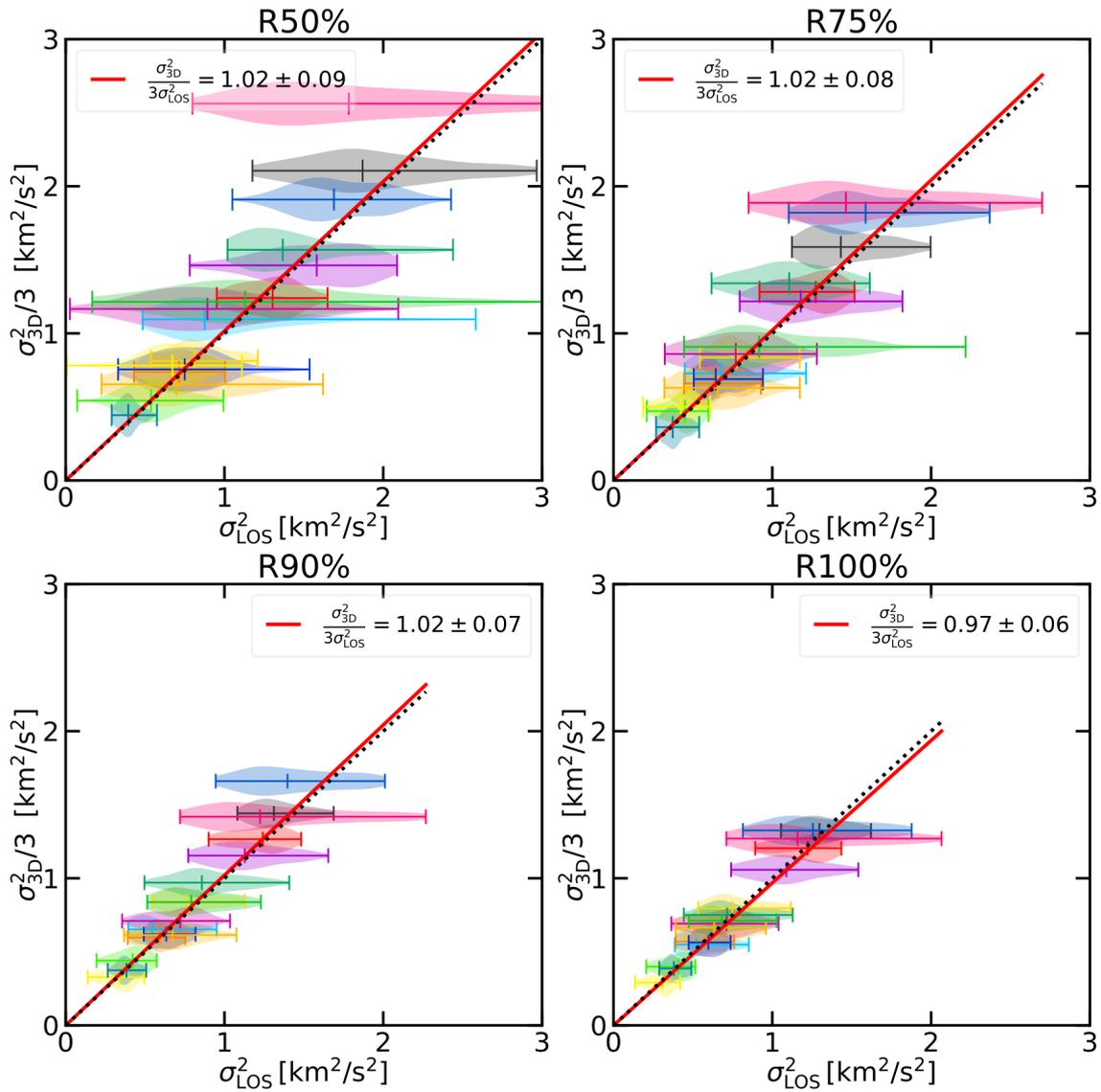


FIGURE 5.3 – Corrélation entre les dispersions des vitesses le long de la ligne de visée et la dispersion des vitesses 3D, estimés à différents rayons englobant 50%, 75%, 90% et 100% de la masse du groupe. La représentation utilisée est la même qu'à la figure 5.2.

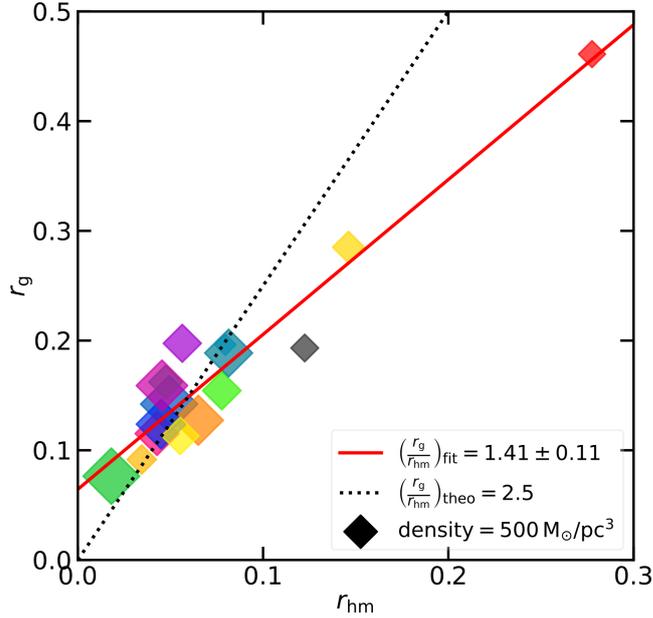


FIGURE 5.4 – Corrélation entre le rayon à mi-masse et le rayon gravitationnel des groupes. La droite rouge montre la régression linéaire réalisée sur les données alors que celle en noire pointillée montre la valeur théorique de 2.5 généralement adoptée pour une majorité de modèles. La taille des symboles est proportionnelle à la densité du groupe correspondant.

faible. Ces derniers sont en effets fortement ségrégués avec seulement une ou deux étoiles centrales représentant plus de la moitié de la masse. Cet effet est bien connu pour fortement augmenter la valeur du facteur η (Fleck et al., 2006). En revanche, les groupes plus massifs et plus riches (en terme de nombre d'étoiles) sont aussi ségrégués mais peuvent posséder plusieurs pics centraux, et ne concentrent pas systématiquement la moitié de leur masse en leur centre. Ils apparaissent donc bien plus étendus et sont globalement encore en train de s'assembler. Ils possèdent donc des distributions de masse encore très diluées (voir aussi leur densité Fig. 5.1). Or ce type de distribution conduisant à des profils de densité moins piqués au centre, à la manière des profils de Elson et al. (1987), dont la densité surfacique Σ suit la loi :

$$\Sigma(r) = \Sigma_0 \left(1 + \frac{r^2}{a^2} \right)^{-\gamma/2}, \quad (5.7)$$

avec a un paramètre d'ajustement et γ déterminant la pente du profil ($\gamma = 4$ correspondant à un profil de Plummer). Pour des profils étendus, lorsque γ varie de 3 à 2, la valeur de η chute brutalement de 10 à presque 0. De telles valeurs de γ sont fréquemment observés pour de jeunes amas massifs (Portegies Zwart et al., 2010; Mackey & Gilmore, 2003; Elson et al., 1987) et pourraient correspondre à nos groupes les plus massifs. Il serait intéressant de tracer de tels profils pour des groupes aussi asymétriques que ceux identifiés dans nos modèles, en extrapolant une luminosité intrinsèque des étoiles grâce à une librairie d'évolution stellaire. Ce travail qui n'a pas pu être réalisé durant cette thèse serait une voie prometteuse d'amélioration de la comparaison entre les modèles GDF et les observations. Pour ce faire, la modularité de la plateforme AMUSE est un atout.

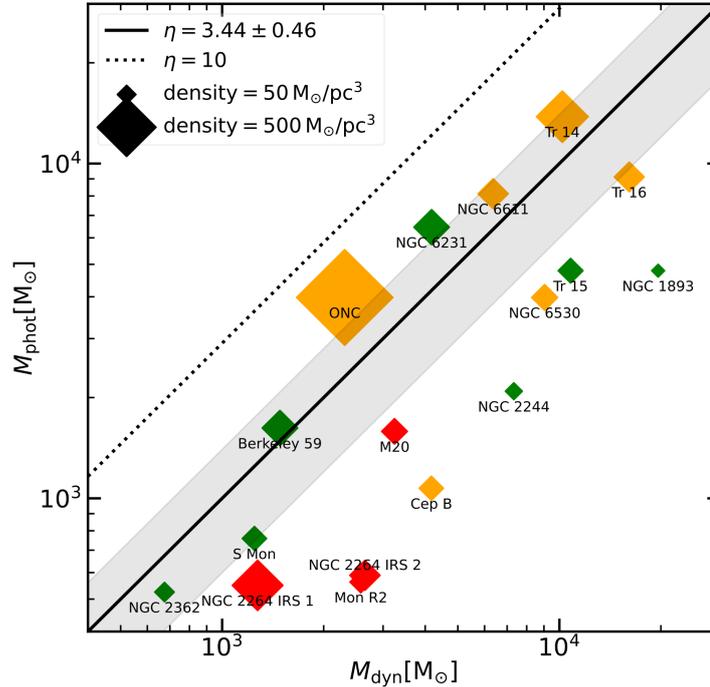


FIGURE 5.5 – Estimation de la masse dynamique des amas et sous-amas identifiés par [Kuhn et al. \(2019\)](#) en fonction de leur masse estimée par photométrie. La ligne noire correspond à une corrélation d'un facteur $\eta = 3.44$ correspondant à la valeur contrainte sur les modèles GDF. La droite en pointillée correspond au facteur de corrélation généralement adopté $\eta = 10$. La taille des marqueurs symbolise la densité des amas. Les couleurs rouge, orange et verte indiquent respectivement si l'amas est encore enfoui dans du gaz, partiellement enfoui ou complètement révélé.

5.1.1 Comparaison avec des groupes identifiés dans *Gaia* DR2

L'avènement de la mission *Gaia* a ravivé l'intérêt sur le sujet de l'identification de sous-structures au sein de régions de formation d'étoiles. [Kuhn et al. \(2019\)](#) ont récemment analysé des amas jeunes du voisinage solaire possédant une grande variété morphologique issue de leur processus de formation. Une partie de ces amas possède plusieurs sous-structures qui sont analysées séparément et permettent une comparaison partielle avec les groupes identifiés dans nos modèles. Grâce à la puissance astrométrique de la mission, les auteurs ont pu mesurer les propriétés morphologiques et cinématiques de certaines de ces structures de manière suffisamment robuste dans le but de questionner leur état dynamique.

La Fig. 5.5 montre la corrélation entre la masse estimée par photométrie et la masse dynamique de ces amas, en utilisant la relation du viriel avec le facteur $\eta = 3.44$, correspondant à la valeur contrainte sur les groupes issus du modèle GDF du chapitre 4. Si les groupes identifiés par [Kuhn et al. \(2019\)](#) ressemblaient aux groupes des modèles GDF, ceux-ci devraient se répartir proche de la droite noire. On observe a contrario, qu'une majorité se situe vers des masses dynamiques plus importantes que la masse visible indiquant un rayon et/ou une dispersion des vitesses trop importante des membres de l'amas. On remarque que c'est en particulier le cas pour des amas de densité de l'ordre de $\sim 50 M_{\odot}/\text{pc}^3$, tel que NGC 1893, NGC 2244 et Cepheus B. Ces derniers montrent des signes clairs d'expansion expliquant ainsi leur position sur la figure. Cela implique par ailleurs,

qu'à l'époque de leur formation ils aient été plus denses, à l'image de Tr 16 qui partage ces mêmes propriétés.

D'autres amas quant à eux, sont plus proches de la droite noire indiquant des propriétés dynamiques similaires aux groupes des modèles GDF. Ils sont aussi majoritairement plus denses. Ce résultat est en accord avec le fait qu'en raison de leur forte densité ils ont déjà eu le temps de se relaxer à l'instar de la nébuleuse d'Orion (ONC) qui arbore une distribution relativement sphérique et concentrée. De plus ils correspondent tous à des régions massives dont l'évolution est probablement le résultat de la coalescence de plusieurs petits amas. Ces derniers pourraient ainsi ressembler aux groupes produits par les modèles GDF. Dorval et al. (2017) à montré que la fusion de ces groupes se produisait de façon hiérarchique ne conduisant pas à de violents mouvements de contraction ni d'expansion en accord avec les observations de Kuhn et al. (2019).

Notons par ailleurs, qu'en faisant l'hypothèse d'une valeur de $\eta = 10$ (ligne noire pointillée), toutes ces régions apparaîtraient sur-viriel ce qui indiquerait une dissociation rapide dans le champ, ce qui ne correspond pas à ce qui est observé. Il apparaît donc que contraindre ce rapport masse/lumière pour des amas et sous-amas jeunes soit crucial dans le but de prédire leur évolution. L'utilisation de modèles pour continuer ce travail sera d'autant plus important que les incertitudes sur l'estimation de la masse photométrique de ces amas restent significatives.

5.2 Cinématique des groupes

Au delà de la structure particulière des groupes produits par les modèles GDF. La Fig. 5.3, montre qu'ils possèdent une cinématique complexe. La dispersion des vitesses de chaque groupe d'un panneau à l'autre, indique globalement qu'elle décroît à mesure que le rayon augmente, signe que les groupes sont partiellement relaxés. Néanmoins, l'extension que couvre chaque distribution due aux multiples ligne de visée, est un indice que la distribution des vitesses au sein des groupes est assez anisotropique. Une mesure de l'anisotropie d'un système peut être mesurée par le paramètre :

$$\beta = 1 - \frac{\sigma_{\tan}^2}{2\sigma_r^2}, \quad (5.8)$$

où σ_{\tan} et σ_r sont respectivement la dispersion tangentielle et radiale des vitesses par rapport au centre de masse du système (attention « radiale » ne veut pas dire ici le long de la ligne de visée). On considère pour des systèmes réels, qu'une valeur de $\beta = 0$ indique une distribution isotrope des vitesses alors que $\beta = 1$ correspond à des orbites uniquement radiales (c.-à-d. vers ou à l'opposé du centre de l'amas). La Fig. 5.6 montre la valeur de ce paramètre calculé pour chaque groupe en fonction de la médiane des vitesses radiales de ces membres. Une corrélation positive entre ces deux mesures (avec $\beta > 0$ et $V_r > 0$) serait le signe de systèmes en expansion nette avec une forte dispersion. Une anti-corrélation indiquerait au contraire des systèmes en effondrement. La plupart des groupes montrent une légère anisotropie radiale avec β légèrement positif. En revanche il n'existe pas de lien clair avec la vitesse radiale moyenne des groupes. On observe seulement pour le groupe ayant la valeur de β la plus élevée qu'il possède aussi la vitesse radiale moyenne la plus élevée indiquant que ce groupe se disperse. Pour les autres, cette anisotropie dénote plus probablement que l'assemblage de ces structures n'est pas terminé.

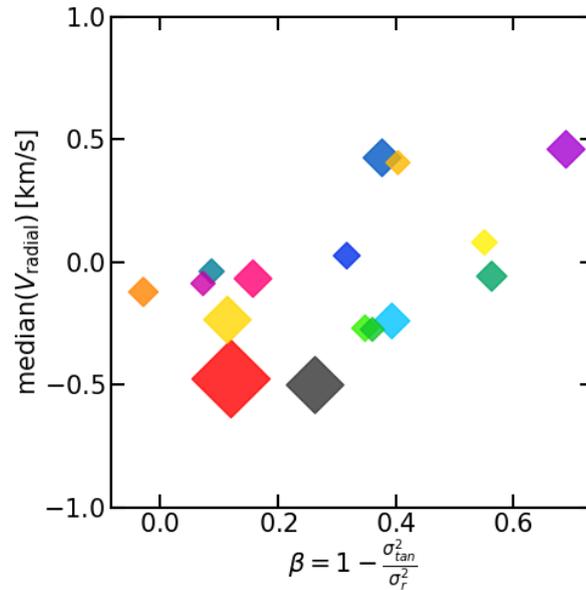


FIGURE 5.6 – Valeur du paramètre β mesurant l’anisotropie des mouvement au sein de chaque groupe en fonction de la vitesse radiale médiane des groupes (vis à vis du centre de masse du groupe).

On peut rappeler aussi que ces modèles n’incluent pas une population de binaires représentative (voir chapitre 6 suivant). La fraction de binaires au sein des groupes est inférieure à 5%. En effet les seules binaires qui existent ont été formées dynamiquement et sont relativement larges. Ainsi leur dispersion de vitesse se compare à celle du système et ne contribue pas à la biaiser significativement. De plus, on observe qu’elles sont encore assez rares au sein des groupes. En effet dans ces environnements denses où les effets collisionnels sont plus marqués, elles sont facilement détruites (ou ne peuvent se former). Certaines binaires serrées pourraient néanmoins rester relativement non-perturbées si elles étaient incluses dans nos modèles. La prise en compte de ce type de binaires pourrait artificiellement augmenter la dispersion des vitesses mesurée. Pour des amas ouverts ayant une fraction de binaires comprise entre 8 et 40%, [Rastello et al. \(2020\)](#) ont estimé une augmentation d’un facteur de 2.5 à 45 de la dispersion des vitesses mesurée par spectroscopie. Néanmoins cet effet est une fonction forte des propriétés des binaires de la population observée et des limites de détection des observations. Dans le cas où les binaires ne sont pas résolues spatialement, des spectres stellaires complexes pourraient indiquer la présence d’un compagnon serré.

Nous revenons dans le chapitre suivant sur la prise en compte des systèmes binaires et multiples au sein des modèles GDF. Leur intégration précise à l’intérieur d’amas denses est importante dans le but d’inférer leur dynamique.

BIBLIOGRAPHIE

- Dorval, J., Boily, C. M., Moraux, E., & Roos, O. 2017, *MNRAS*, 465, 2198
- Elson, R. A. W., Fall, S. M., & Freeman, K. C. 1987, *ApJ*, 323, 54
- Fleck, J. J., Boily, C. M., Lançon, A., & Deiters, S. 2006, *MNRAS*, 369, 1392
- Kuhn, M. A., Hillenbrand, L. A., Sills, A., Feigelson, E. D., & Getman, K. V. 2019, *ApJ*, 870, 32
- Mackey, A. D. & Gilmore, G. F. 2003, *MNRAS*, 338, 85
- Parker, R. J. & Wright, N. J. 2016, *MNRAS*, 457, 3430
- Portegies Zwart, S. F., McMillan, S. L. W., & Gieles, M. 2010, *ARA&A*, 48, 431
- Rastello, S., Carraro, G., & Capuzzo-Dolcetta, R. 2020, *ApJ*, 896, 152
- Spitzer, L. 1987, Dynamical evolution of globular clusters (Princeton Series in Astrophysics)
- Zamora-Avilés, M., Ballesteros-Paredes, J., & Hartmann, L. W. 2017, *MNRAS*, 472, 647

ÉVOLUTION DES POPULATIONS DE BINAIRES AU SEIN D'ENVIRONNEMENTS DENSES

Dans ce chapitre, nous nous intéresserons aux petites structures que représentent les systèmes multiples. Au sein des amas d'étoiles, ces structures élémentaires sont un facteur déterminant des processus dynamiques à l'œuvre car elles sont le lieu d'interactions fortes. Nous détaillerons une méthode numérique implémentée sous AMUSE pour traiter de manière précise une simulation aux N -corps contenant des systèmes binaires. Nous étudierons aussi la population de ces systèmes créés au sein des modèles GDF.

6.1 Observations

Les binaires sont des systèmes de deux étoiles liées par leur gravité. Un couple d'étoiles est lié si leur énergie de liaison E est négative :

$$E = \frac{1}{2}\mu\Delta v^2 - \frac{GM\mu}{\Delta r} < 0, \quad (6.1)$$

où Δv et Δr sont respectivement la vitesse et la distance relative entre les deux composantes ainsi que $M = m_1 + m_2$ et $\mu = \frac{m_1 m_2}{M}$ la masse totale et réduite du système. Les deux corps suivent alors des trajectoires elliptiques selon les lois de Kepler. Ces dernières possèdent une période de révolution P fonction du demi-grand axe majeur a de l'orbite qu'elles décrivent :

$$P = \frac{2\pi a^{\frac{3}{2}}}{\sqrt{GM}} \quad (6.2)$$

En l'absence de couple de forces extérieures, ce système est stable. En revanche, nombre d'interactions peuvent conduire à une modification du budget énergétique de ce système et modifier en conséquence les paramètres orbitaux. C'est ainsi que l'étude des binaires est d'un intérêt particulier car elles offrent aux observateurs de nombreux traceurs des phénomènes en jeu dans les environnements qu'elles traversent. Néanmoins leur observation reste toujours un challenge, car la distance séparant les deux composantes d'un système binaire se compte en unités astronomiques. De plus, il existe des systèmes où une voire plusieurs composantes sont elles mêmes des systèmes binaires. Ces systèmes hiérarchiques où les différentes composantes s'étendent sur des échelles de temps et de distances distinctes complexifient encore leur description.

On caractérise ainsi la proportion de systèmes multiples par deux mesures. La fraction de systèmes multiples (MF, pour *Multiplicity Fraction*) ne fait pas la distinction entre les systèmes de différents ordres alors que la fraction de compagnon CF, pour *Companion Fraction*) prend en compte le nombre de composantes au sein de chaque système. Sachant s le nombre d'étoiles seules, b celui de binaires, t celui de triples au sein d'un

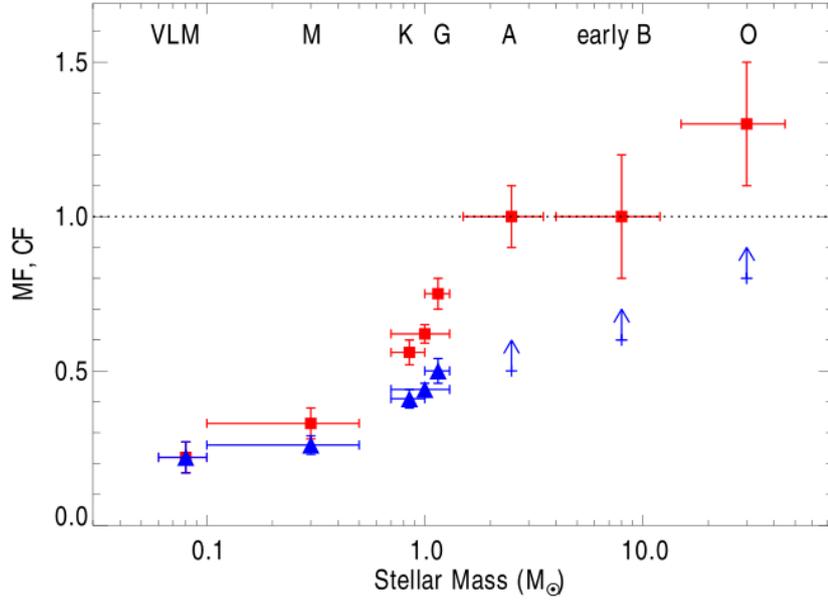


FIGURE 6.1 – Proportion de systèmes multiples (triangles bleus) et de compagnons (carrés rouges) en fonction de la masse des étoiles de la séquence principale du champ galactique. Les barres d'erreurs horizontales indiquent la taille de l'échantillon en masse. Figure extraite de [Duchêne & Kraus \(2013\)](#).

échantillon :

$$\text{MF} = \frac{b + t}{s + b + t} \quad (6.3)$$

$$\text{CF} = \frac{b + 2t}{s + b + t} \quad (6.4)$$

On note que la seconde définition peut conduire à un taux supérieur à 100%. Si la deuxième mesure offre une description plus précise des systèmes multiples, elle est aussi plus difficile à mesurer et sujette à des variations lorsque, par exemple, une binaire est reclassée en système triple en raison d'une nouvelle observation plus précise. Ce qui n'est pas le cas de la MF. Cette dernière permet aussi de caractériser simplement un système multiple par la masse de sa composante la plus massive, c.-à-d. sa primaire. On préfère donc en général, la mesure de la MF pour sa simplicité et permettre une comparaison entre différentes populations, nous ferons aussi ce choix tout au long de ce chapitre. Les observations dans notre galaxie montrent que la proportion de systèmes multiples augmente avec la masse de la primaire. Par exemple, la MF varie de 20% pour les naines brunes à près de 100% pour les étoiles de types OB ([Duchêne & Kraus, 2013](#)). La Fig. 6.1 montre la dépendance de la MF et la CF avec la masse des étoiles sur la séquence principale.

Ces valeurs sont bien contraintes sur les populations d'étoiles du champ galactique et de la séquence principale. Ce n'est pas le cas au sein des régions de formation d'étoiles. Bien que plus difficile à sonder, on observe dans les populations jeunes autant voire plus de systèmes multiples que dans le champ ([Mathieu, 1994](#); [Duchêne & Kraus, 2013](#)). Il a été proposé une séquence évolutive où les étoiles naissent avec un fort taux de binaire, progressivement détruites au fur et à mesure qu'elles se dissolvaient dans le champ. Néanmoins l'ob-

servation de certaines régions jeunes comme IC 348 ou Orion montre des fractions de binaires relativement basse (Duchêne et al., 1999). Le taux de binaires semble alors déterminé très tôt dans la vie des étoiles et semble pouvoir varier de manière importante à l'issue du processus de formation. En faisant l'hypothèse d'une MF de 100% à la naissance, Marks & Kroupa (2012) ont montré que les violents processus collisionnels dans les amas denses pouvaient conduire à une MF similaire au champ en quelques M.a.. La densité de l'environnement natal semble être un critère déterminant. Les régions peu denses comportant un nombre important de systèmes multiples alors que des amas plus denses comportent une MF comparable au champ. Ainsi, il n'est pas encore défini si cette anti-corrélation est la marque de processus de formation des étoiles différents selon les régions, ou n'est que le résultat d'une destruction plus rapide des binaires dans les amas denses.

Les simulations hydrodynamiques sont capables de reproduire des résultats en accord avec les observations, concernant les étoiles peu massives (Bate, 2012). Néanmoins elles n'expliquent pas les grands écarts observés dans la distribution de séparation de ces systèmes. Cela semble indiquer différents processus de formation de ces systèmes (voir p. ex. Tohline, 2002). De cette manière, les binaires sont aussi un outil pour sonder l'environnement dans lequel elles se forment. Dans ce contexte, les simulations permettent de mieux contraindre les processus modifiant les caractéristiques des populations de binaires au sein de leur environnement natal.

6.2 Simulations

On a vu au chapitre 2 que l'intégration d'un système binaire représentait un défi pour les intégrateurs aux N -corps. En effet les variations de l'accélération subies par des particules en interactions fortes ou au sein d'une binaire excentrique nécessitent une intégration utilisant un pas de temps extrêmement petit comparé au temps d'évolution globale du système. Cette limitation a conduit à l'émergence de techniques pour pallier cette difficulté.

La plus largement utilisée consiste à adopter un pas de temps différent pour chacune des particules du système en fonction de l'accélération qu'elles subissent. Ainsi le calcul de l'accélération n'est répété que pour les particules soumises à de fortes variations de leur vecteur vitesse, typiquement dans des environnements denses. De plus, pour permettre à toutes les particules de se synchroniser à intervalles réguliers, ces pas de temps sont calculés pour être commensurables. Ainsi on attribue généralement un pas de temps égal à une puissance de 2 du plus petit choisi. Cette méthode, sous le nom de *blocked timestep* (McMillan, 1986; McMillan & Aarseth, 1993), est bien adaptée pour améliorer la précision de l'intégration lors d'interactions fortes, mais brèves car elle focalise la précision de l'intégration là où c'est nécessaire. Néanmoins, en présence de binaires stables, l'intégrateur est sans cesse assujéti à résoudre le détail de leurs orbites respectives, ce qui ralentit énormément le calcul.

Pour traiter cette deuxième difficulté, l'approche généralement utilisée consiste à découpler l'intégration des binaires du reste du système, puisqu'elles évoluent sur des échelles de temps et de distance, séparées de plusieurs ordres de grandeur. De plus, si le système binaire n'est pas perturbé, le calcul de la trajectoire de ses composantes est connu analytiquement. Les gains potentiels en temps de calcul sont donc importants.

La plupart des intégrateurs intègrent une méthode de régularisation à deux corps. Grâce à un changement de coordonnées adéquat, il est possible d'ôter la singularité posée par le calcul de l'accélération lorsque la distance entre deux particules tend vers 0. Il n'est donc plus nécessaire de décroître de manière prohibitive le pas de temps d'intégration. De plus, il est possible de prendre en compte des perturbations extérieures. Cette méthode est en particulier utilisée via la procédure de Kustaanheimo-Stiefel au sein du code NBODY6 (Aarseth, 1999; Mikkola & Aarseth, 1993). Toutefois, elle est difficile à implémenter et conduit à complexifier le schéma d'intégration.

Une méthode alternative consiste à identifier et isoler de manière plus stricte les binaires. Ces dernières sont intégrées au sein du reste du système dans l'approximation de leur centre de masse. Leur orbite n'est précisément recalculée qu'en cas d'interactions fortes. Cette approche popularisée au sein des codes Monte Carlo (Tanikawa & Fukushige, 2009), permet d'éviter l'intégration coûteuse pour chacun de ces systèmes, en ne calculant qu'une seule fois leurs paramètres orbitaux. Ainsi plutôt que de changer de coordonnées, cette méthode nécessite une autre stratégie algorithmique. Cette approche est particulièrement adaptée à la plateforme AMUSE qui utilise un deuxième code spécifique avec l'aide un module spécialement dédié, `Multiples`.

6.2.1 Approche AMUSienne : le module `Multiples`

Le couplage entre différents codes est au cœur de la philosophie d'AMUSE. Le script guidant une simulation doit pouvoir interagir facilement avec les différents codes utilisés. Cette gestion peut se faire directement dans le script principal ou en utilisant un module dédié comme le module `Multiples` pour les interactions fortes. Tous les codes que comportent AMUSE intègrent une gestion des événements permettant de stopper l'intégration et de rendre la main au script principal. Cela peut concerner de potentiels crashes du calcul mais aussi des événements particuliers utiles à la compréhension de l'évolution du système. On utilise cette fonctionnalité pour effectuer un traitement particulier de l'événement considéré. Ceci peut concerner l'explosion d'une supernova dans le cas d'un code d'évolution stellaire ou d'une collision entre deux particules dans le cas N -corps. C'est ce dernier cas qui est exploité par `Multiples` qui offre l'exemple d'un traitement spécifique assez poussé.

Le module `Multiples` se présente comme une classe PYTHON et se charge de donner la main à différents codes. La stratégie du module `Multiples` consiste à en coupler 3 résolvant chacun des situations spécifiques :

- Un premier intègre le système à grande échelle et se charge de l'évolution des étoiles seules et des macro-particules représentant le centre de masse des systèmes multiples. On parlera du code de « haut-niveau ».
- Un deuxième intègre les interactions à petite échelle lorsque nécessaire, typiquement lors d'interaction forte (code à « petit- n »).
- Enfin le dernier fonctionne de concert avec le précédent, en fournissant la solution analytique d'orbites képlériennes.

Mis à part le dernier, chacun des codes à grande et à petite échelles peuvent être choisis par l'utilisateur. On préférera donc pour le code de « haut-niveau », un code pouvant intégrer l'ensemble du système de manière

rapide et précise car il supportera la majorité de l'intégration. Le code à « petit- n » est quant à lui plus spécifique et plus intimement lié au fonctionnement du module `Multiple`. Ce dernier est d'ailleurs configuré pour travailler avec le code `smallN` (Hut et al., 1995; McMillan & Hut, 1996; Portegies Zwart et al., 1999). La transition entre l'intégration globale et une interaction à petite échelle est réalisée au sein du code de « haut-niveau » lorsqu'une collision est détectée. Pour cela, il est assigné à chacune des particules du système un rayon dynamique. Ce rayon correspond à une section efficace d'interaction avec les autres particules du système. De manière plus pratique, lorsque la distance entre deux particules devient inférieure à la somme de leurs rayons respectifs, le code détectera une collision. À la suite de cela, le module `Multiple` prend en considération les particules impliquées. Notons que ces particules peuvent elles mêmes, être des macro-particules représentant des systèmes multiples. Dans cette situation `Multiple` se charge de séparer chaque composante en fonction des paramètres orbitaux qu'il a sauvegardé. Ensuite, avec l'aide du code à « petit- n », il détermine le résultat d'une telle interaction. Ce dernier n'est en rien trivial, notamment lorsque plus de deux particules sont impliquées. C'est en partie pour cette raison que le module est configuré par défaut avec le code `smallN` qui traite le problème par une méthode statistique (on parle de *scattering experiment* McMillan & Hut, 1996). La fin de l'interaction est actée lorsque toutes les composantes forment un système multiple stable (selon les critères de Mardling, 2008) ou lorsqu'elles se sont suffisamment éloignées les unes des autres. Ce dernier critère permet d'éviter la création de systèmes multiples à trop fortes séparations pour lesquelles l'approximation du centre de masse ou d'un temps d'interaction instantané ne tiendrait plus. Dans le cas où un système multiple s'est formé, ses paramètres orbitaux sont sauvegardés et il est remplacé par une macro-particule au centre de masse du système. On lui assigne aussi un rayon dynamique correspondant à deux fois le semi-grand axe du système. Enfin, le module `Multiple` rend la main à l'intégration de « haut-niveau » qui peut continuer l'évolution du système dans son ensemble. Les différentes étapes de cette procédure sont schématisées dans la Fig. 6.2.

Cette procédure possède l'énorme avantage de soulager l'intégrateur de « haut-niveau » qui ne peut suivre l'intégration de l'ensemble des systèmes multiples à la fois de façon rapide et précise. Néanmoins elle se fait au détriment d'autres approximations qu'il est utile de lister :

1. Les interactions faisant intervenir le module `Multiple` se font de manière instantanée. En effet, la procédure stoppe l'intégration du système global le temps que l'interaction soit résolue. Cette approximation est en général valide car ne dure qu'un très bref instant (de l'ordre de $1/N$ temps dynamique pour une simple interactions à deux corps). En revanche lorsqu'un système complexe et hiérarchique est impliqué, il est possible que cela produise une série d'interactions chaotiques pouvant durer un temps non-négligeable (Hut, 1993; Geller & Leigh, 2015). Cette dérive est limitée par `Multiple` qui ne considère pas les systèmes trop larges ou résonnant comme stables.
2. Lors de l'intégration de « haut-niveau », les composantes des macro-particules représentant les centres de masse des systèmes multiples ne sont pas intégrés. Ainsi la configuration précise (comme la phase ou l'inclinaison de l'orbite) de ces systèmes est perdue.
3. Enfin, l'approximation du centre de masse permet seulement l'inclusion de certaines perturbations à courtes distances par les étoiles voisines.

Cette dernière approximation mérite une attention particulière car le module `Multiple` inclut une

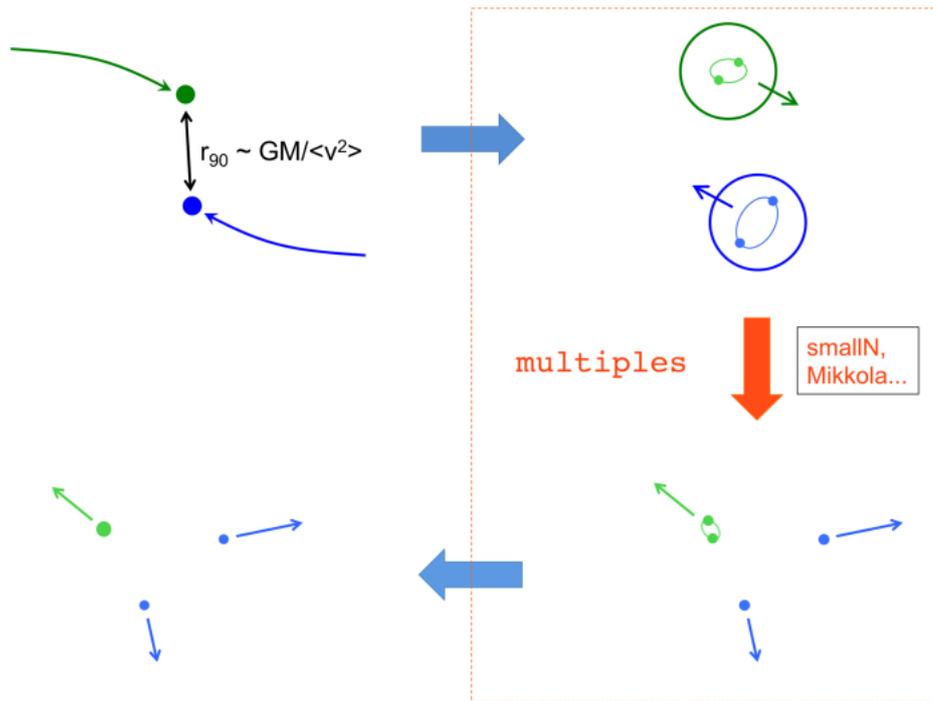


FIGURE 6.2 – Principe de fonctionnement de la procédure du module `Multiple` : En partant d'en haut à gauche et en suivant les flèches larges, le code de « haut-niveau » détecte une interaction forte entre deux particules (ici la distance entre les deux particules devient inférieure au paramètre d'impact correspondant à une déflexion à 90°). Il s'arrête et transmet les particules au module `Multiple` (dans le cadre rouge, en haut à droite). Ce dernier réalise quelques tests et transmet les particules au code à « petit- n ». Si, comme sur la figure, les particules en jeu sont des systèmes multiples, leurs composantes sont séparées et replacées en fonction des paramètres orbitaux sauvegardés par le module. Le code à « petit- n » intègre précisément le résultat d'une telle interaction jusqu'à ce que chacune des composantes soient déliées (en bas à droite). Dans le cas où un système stable est formé (ici, les particules vertes), `Multiple` crée une macro-particule représentant le centre masse du système. Enfin toutes les particules sont rendues au code de « haut-niveau » qui peut reprendre l'intégration globale du système (en bas à gauche). Figure extraite de [Portegies Zwart & McMillan \(2018\)](#).

procédure permettant de traiter ces perturbations. Une étude réalisée durant cette thèse explorant ces fonctionnalités a permis de relever une erreur dans le schéma algorithmique. Ainsi, nous détaillons cette procédure ci-dessous.

6.2.2 Traitement des perturbations par le module `Multiple`

Le principe de base du fonctionnement du module `Multiple` consiste à considérer les systèmes multiples comme isolées du système. Or dans les systèmes denses que sont les amas d'étoiles, cette approximation peut être mise en défaut. Ainsi une paramétrisation du module permet d'ajuster son fonctionnement selon les situations rencontrées.

Lorsqu'une interaction forte est détectée, le module `Multiple` vérifie en premier lieu si les particules concernées ne subissent pas de perturbations du système global. Ceci est évalué par un critère de densité.

Ainsi, si les deux particules en interactions sont de masse m_1, m_2 et séparées d'une distance a , l'interaction est considérée comme perturbée par une troisième particule de masse m_3 distante de d du centre de masse des deux autres, si :

$$\gamma_p \frac{m_1 + m_2}{2a^3} < \frac{m_3}{d^3}, \quad (6.5)$$

Le facteur γ_p fixe un seuil de densité relatif à partir duquel le module considère l'interaction comme perturbée ou non. La valeur par défaut choisi est de l'ordre de quelques pourcents ($\gamma_p = 0.02$).

À partir de cette étape, le module `Multiple` permet deux traitements différents. Le premier qui est celui par défaut consiste à considérer que l'environnement joue un rôle important au moment de cette intégration qui ne peut donc être réalisé de manière séparée. Ainsi, le module rend la main immédiatement au code de « haut-niveau » qui aura la charge de cette intégration. Cette méthode a l'avantage d'éviter une approximation qui serait difficile d'évaluer. Néanmoins, elle empêche le module `Multiple` de jouer pleinement son rôle. Le deuxième traitement consiste au contraire à prendre en compte les particules perturbant le système dans l'intégration à petite échelle. Ainsi toutes les particules ne satisfaisant pas au critère de l'Eq. 6.5 sont ajoutées comme participant à l'interaction. Dans cette situation, plus de calcul est demandé au code à « petit- n ». Cette approche permet un potentiel gain en précision, mais il est aussi possible que le calcul ne converge pas. C'est dans ce dernier cas que nous avons identifié un problème algorithmique du module `Multiple`.

Le crash d'un code numérique n'est pas exceptionnel, d'autant plus que la non-convergence du code à « petit- n » dans notre cas, conduisait le module `Multiple` à rendre la main au code de « haut-niveau » pour réaliser l'intégration moyennant une précision dégradée (cette situation revenait en fait à ré-adopter le traitement par défaut). En revanche, le retour à l'étape initiale de la détection de l'interaction n'était pas réalisé correctement. En réalité, nous avons observé que certaines particules étaient tout simplement effacées du système! Après de nombreux tests et inspections du code, nous avons réalisé que les particules en question n'étaient pas les particules qui avaient déclenchées la collision, mais les particules perturbatrices ajoutées a posteriori au code à « petit- n » pour prendre en compte leur impact.

La détection de cette mauvaise gestion des particules a été faite lors d'une étude visant à comparer l'évolution de populations de binaires sous différents environnements. Ce travail effectué au début de ce projet de thèse visait à explorer l'évolution des paramètres des binaires au sein des surdensités créées par les modèles GDF. Contrairement à des conditions jugées plus classiques – consistant en de petits amas possédant p. ex. un profile de King (King, 1962) virialisé – la relaxation des surdensités issues d'amas GDF conduisait à un nombre faible mais non négligeable d'interactions fortes conduisant à la non-convergence du code à « petit- n ». Certaines particules étaient ainsi perdues durant l'évolution du système. Puisque les surdensités considérées ne comportaient qu'une centaine d'étoiles, cette erreur devenait significative.

Cette erreur a été l'objet de discussion avec l'équipe d'AMUSE. En raison de la complexité du traitement du module `Multiple` et le fait que les particules effacées n'étaient pas directement impliquées dans les interactions qui conduisaient à leur éviction, le problème n'a pas pu être réglé de suite et cette étude a donc été avortée. Ce problème fait resurgir un problème de stabilité récurrent avec AMUSE dont le développement est toujours actif. Une attention particulière doit donc être prêtée à chaque étape modifiant le fonctionnement par défaut des codes, et des tests scrupuleux sont absolument nécessaires dans le but de vérifier le fonction-

nement attendu. Il est aussi fortement recommandé de vérifier les mises à jour et les problèmes résolus par la communauté sur le dépôt *github*¹.

Cependant, les discussions engagées avec les concepteurs ont permis de mieux appréhender le fonctionnement globale du module `Multiple`. Notre utilisation a permis de nombreux autres développements. Nous l’avons aussi utilisé pour construire de nouveaux modèles GDF et son traitement particulier des systèmes multiples pourrait permettre d’améliorer ces modèles. Le script permettant de créer un modèle GDF grâce à ce module est donné en annexe [A.3.2](#)

6.3 Applications

Les modèles GDF que nous avons étudiés jusqu’ici comportent peu de binaires. En effet nous n’avons pas inclus de population de binaires dans les conditions initiales. Cela n’empêche pas certains systèmes de se former par capture durant la phase d’expansion. Nous montrons sur la Fig. 6.3, la fraction de systèmes multiples au sein d’un de nos modèles de 15 000 étoiles en fonction de la masse de la primaire. Nous comparons ces résultats avec les données observationnelles issues de [Duchêne & Kraus \(2013\)](#) ainsi que celles issues d’une simulation hydrodynamique de [Wall et al. \(2019\)](#). Cette dernière a été récemment réalisée par les concepteurs d’AMUSE dans le but d’intégrer à la plateforme le code de magnéto-hydrodynamique FLASH ([Fryxell et al., 2000](#)). Elle reprend les conditions traditionnellement adoptées de simulations de formation d’étoiles par l’effondrement d’un nuage moléculaire dense et turbulent (du type de [Bate, 2012](#)) avec ici la prise en compte d’un champ magnétique initialement uni-directionnel et uniforme. Les étoiles sont artificiellement formées par des *sink particles* dont la masse est fixée pour reproduire l’IMF. L’utilisation du module `Multiple` dans leur calcul permet une comparaison intéressante avec nos modèles. De plus, la résolution limitée de leur calcul ne permet pas de former des binaires primordiales par fragmentation du cœur pré-stellaire (contrairement à [Bate, 2012](#)). La population de systèmes multiples qu’ils analysent est ainsi formée dynamiquement au sein de l’amas dense en train de se former.

Les résultats des deux méthodes suivent la même tendance, celle d’une augmentation de la fraction des systèmes multiples avec la masse de la primaire. On observe toutefois un écart variable avec les observations. Cet écart est moins grand pour les modèles GDF. Néanmoins, ces derniers sont à un stade moins évolués d’à peine un temps dynamique. En étudiant la relaxation des modèles GDF, ([Dorval et al., 2017](#)) a déjà montré qu’environ 20% des binaires pouvaient être détruites, conduisant à une baisse de 5 à 20% de la MF. De cette manière les deux simulations pourraient mieux s’accorder sur la fraction de binaires formées dynamiquement.

L’étude des paramètres orbitaux de ces deux populations, permet une comparaison plus poussée. Ces derniers sont affichés sur les Fig. 6.4 et 6.5. On retrouve de nouveau une bonne corrélation concernant la masse de la primaire dont la majorité se regroupe autour de $1 M_{\odot}$. La distribution des semi-grands axes culmine autour de 10^3 - 10^4 U.A., s’étendant même pour la simulation de [Wall et al. \(2019\)](#), à des valeurs de plus de 1 parsec. Ces valeurs extrêmes ne sont possibles que pour des systèmes qui ont été éjectés de l’amas central et sont très faiblement liés. Cela confirme que les systèmes formés dynamiquement correspondent à des systèmes plutôt larges et ne représentent pas la majorité des systèmes binaires observés. En effet la distribution des semi-grands

1. <https://github.com/amusecode/amuse/issues>

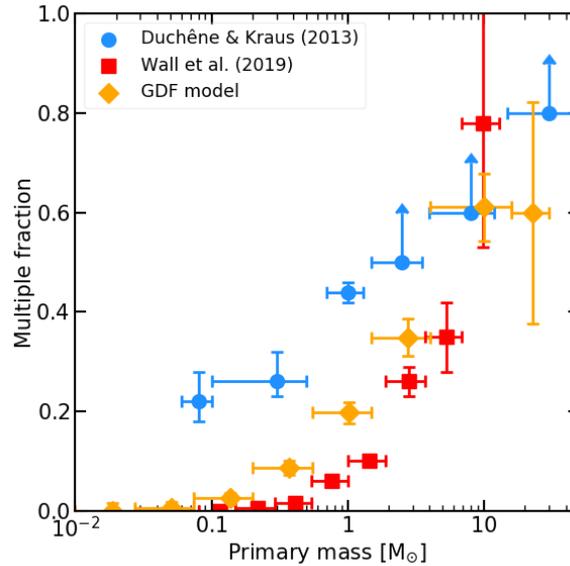


FIGURE 6.3 – Fraction de systèmes multiples au sein d’un modèle GDF de 15 000 particules (en orange), et extrait des simulations hydrodynamiques de Wall et al. (2019) (en rouge). Les barres d’erreurs horizontales indiquent la tranche en masse considérée tandis que les barres d’erreurs verticales correspondent aux fluctuations poissonniennes en $N^{-0.5}$. Les données observationnelles issus de Duchêne & Kraus (2013) sont symbolisés en bleu pour comparaison.

axes est souvent ajustée par une loi log-normale culminant à des séparations de l’ordre de 100 U.A. (Raghavan et al., 2010).

Enfin, la distribution des excentricités montre une surreprésentation de valeurs supérieures à 0.9 avec une moyenne d’environ 0.8 indiquant un processus de capture particulier au sein des modèles GDF. Cette valeur moyenne est en effet supérieure à la valeur ~ 0.66 obtenue pour une distribution « thermique » des excentricités ($f(e) = 2e$ Kroupa, 2008, correspondant à une répartition uniforme d’énergie; notons que cette distribution n’est généralement pas observée Raghavan et al. 2010; Duchêne & Kraus 2013).

Les caractéristiques particulières de ces systèmes, formés au sein de ces simulations montrent des différences marquantes avec les observations, avec notamment la sous-représentation de systèmes de faibles masses et de binaires plus serrées. Ces résultats confirment ainsi la nécessité d’un mode de formation de ces systèmes pendant la phase pré-stellaire. La prise en compte d’une population de binaires primordiales permettrait de mieux comprendre leur évolution au sein de jeunes amas. En ce qui concerne les simulations hydrodynamiques, cela nécessiterait un raffinement de la résolution qui est difficilement réalisable. Notons qu’une méthode a été récemment développée pour la simulation de Wall et al. (2019) via une méthode d’inclusion d’une population de binaires ajustée sur les observations (Courmoyer-Cloutier et al., 2020). Une approche similaire a déjà été testée par Dorval et al. (2017). Dans ce dernier cas, cette procédure d’introduction de binaires à la fin de la fragmentation est quelque peu artificielle car elle ne prend pas en compte l’environnement dans lequel la binaire introduite est immergée. Elle pourrait donc être améliorée, en incluant cette population dès le début de l’expansion du modèle GDF grâce au module `Multiple`.

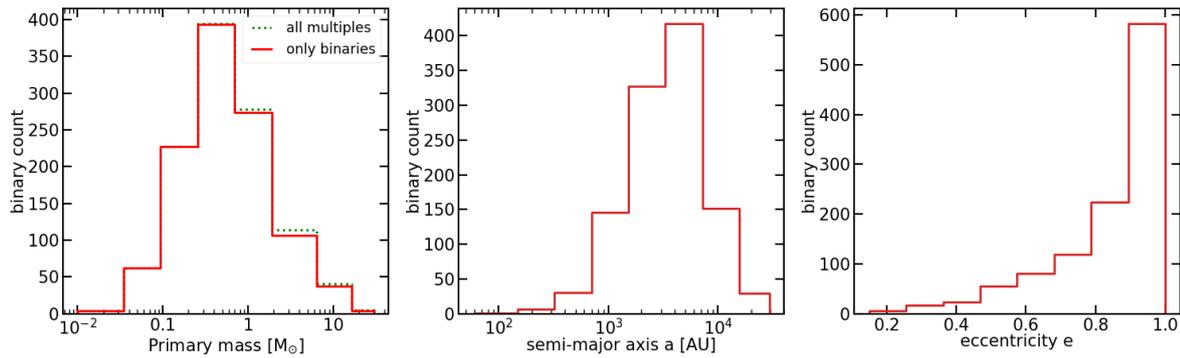


FIGURE 6.4 – Propriétés des systèmes multiples au sein d'un modèle GDF. De droite à gauche : distribution de la masse de la primaire (les systèmes multiples comportant plus de deux composantes sont symbolisés par la ligne verte pontillée), du semi-grand axe majeur et de l'excentricité des systèmes binaires.

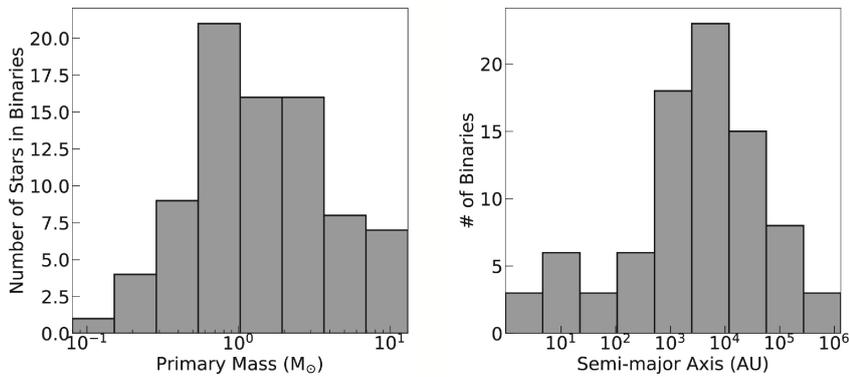


FIGURE 6.5 – Propriétés des systèmes multiples formées au sein de la simulation de [Wall et al. \(2019\)](#).

BIBLIOGRAPHIE

- Aarseth, S. J. 1999, *PASP*, **111**, 1333
- Bate, M. R. 2012, *MNRAS*, **419**, 3115
- Cournoyer-Cloutier, C., Tran, A., Lewis, S., et al. 2020, *arXiv e-prints*, [arXiv :2011.06105](#)
- Dorval, J., Boily, C. M., Moraux, E., & Roos, O. 2017, *MNRAS*, **465**, 2198
- Duchêne, G., Bouvier, J., & Simon, T. 1999, *A&A*, **343**, 831
- Duchêne, G. & Kraus, A. 2013, *ARA&A*, **51**, 269
- Fryxell, B., Olson, K., Ricker, P., et al. 2000, *ApJS*, **131**, 273
- Geller, A. M. & Leigh, N. W. C. 2015, *ApJ*, **808**, L25
- Hut, P. 1993, *ApJ*, **403**, 256
- Hut, P., Makino, J., & McMillan, S. 1995, *ApJ*, **443**, L93
- King, I. 1962, *AJ*, **67**, 471
- Kroupa, P. 2008, Initial Conditions for Star Clusters, ed. S. J. Aarseth, C. A. Tout, & R. A. Mardling, Vol. 760 (Springer Verlag), 181
- Mardling, R. A. 2008, in *Dynamical Evolution of Dense Stellar Systems*, ed. E. Vesperini, M. Giersz, & A. Sills, Vol. 246, 199–208
- Marks, M. & Kroupa, P. 2012, *A&A*, **543**, A8
- Mathieu, R. D. 1994, *ARA&A*, **32**, 465
- McMillan, S. L. W. 1986, The Vectorization of Small-N Integrators, ed. P. Hut & S. L. W. McMillan, Vol. 267, 156
- McMillan, S. L. W. & Aarseth, S. J. 1993, *ApJ*, **414**, 200
- McMillan, S. L. W. & Hut, P. 1996, *ApJ*, **467**, 348
- Mikkola, S. & Aarseth, S. J. 1993, *Celestial Mechanics and Dynamical Astronomy*, **57**, 439
- Portegies Zwart, S. & McMillan, S. 2018, *Astrophysical Recipes : the art of AMUSE*, 2514-3433 (AAS IOP Astronomy publishing)

BIBLIOGRAPHIE

Portegies Zwart, S. F., Makino, J., McMillan, S. L. W., & Hut, P. 1999, *A&A*, 348, 117

Raghavan, D., McAlister, H. A., Henry, T. J., et al. 2010, *ApJS*, 190, 1

Tanikawa, A. & Fukushige, T. 2009, *PASJ*, 61, 721

Tohline, J. E. 2002, *ARA&A*, 40, 349

Wall, J. E., McMillan, S. L. W., Mac Low, M.-M., Klessen, R. S., & Portegies Zwart, S. 2019, *ApJ*, 887, 62

CONCLUSION ET PERSPECTIVES

Nous avons montré dans les précédents chapitres l'intérêt d'étudier via une approche théorique les structures primordiales issues des régions de formation d'étoiles. Que ce soit les petits groupes d'étoiles considérés comme les briques élémentaires de la formation d'étoiles ou les systèmes binaires et multiples impactant directement les propriétés dynamiques et leur évolution de ses groupes sur-denses. Nous détaillons dans ce chapitre quelques développements futurs dont ce travail de thèse pourrait servir de base. L'utilisation de la plateforme AMUSE promet de complexifier cette approche en prenant en compte par exemple l'effet du gaz omniprésent au sein de ces régions ou encore des effets galactiques permettant aux simulations de régions de formation d'étoiles de s'ouvrir à une vision plus globale.

7.1 Incomplétude

L'incomplétude des observations est un problème récurrent en astronomie. Elle est la principale limite à la comparaison d'amas issus d'environnements différents ainsi qu'avec des modèles théoriques. Elle est aussi particulièrement marquée dans l'identification d'amas jeunes car l'assemblage de ces derniers a lieu en seulement quelques millions d'années et donc, à des stades encore enfouis. Dans ces conditions, l'estimation de leur propriétés est généralement moins robuste.

Il existe deux effets combinés limitant les observations. Le premier correspond à la limite de sensibilité des instruments. Les étoiles les plus faiblement lumineuses ne sont pas détectées, et ce d'autant plus que la région considérée est lointaine. Le deuxième correspond à l'obscurcissement dû à la poussière présente le long de la ligne de visée. Le premier de ces phénomènes a été partiellement exploré au chapitre 4 en retirant au sein des groupes analysés les étoiles les moins massives, qui sont au premier ordre les moins lumineuses. Nous n'avons néanmoins pas remis en cause la détection des groupes elle-même. Les groupes des modèles GDF sont déjà ségrégués en masse. On pourrait donc s'attendre à ce que les surdensités de la région où se trouvent les étoiles les plus lumineuses, apparaissent de manière plus marquée. Cependant, quelques tests tentant d'évaluer si les mêmes groupes étaient identifiés pour différentes valeurs de coupure en masse, ne confirment pas de manière claire cette tendance. En effet, la ségrégation en masse de ces groupes n'est pas parfaite et des effets antagonistes sont à l'œuvre. D'une valeur de coupure à l'autre, deux groupes différents peuvent apparaître comme faisant partie de la même surdensité, tout comme un même groupe peut finalement être scindé en plusieurs parties. Ces effets non-linéaires montrent de nouveau l'intérêt des modèles GDF dans l'étude des biais introduits par les algorithmes de détection.

La deuxième limitation due à l'extinction le long de la ligne de visée va aussi s'ajouter à la première. Il existe des méthodes permettant de corriger partiellement les observations de cet effet. Une technique uti-

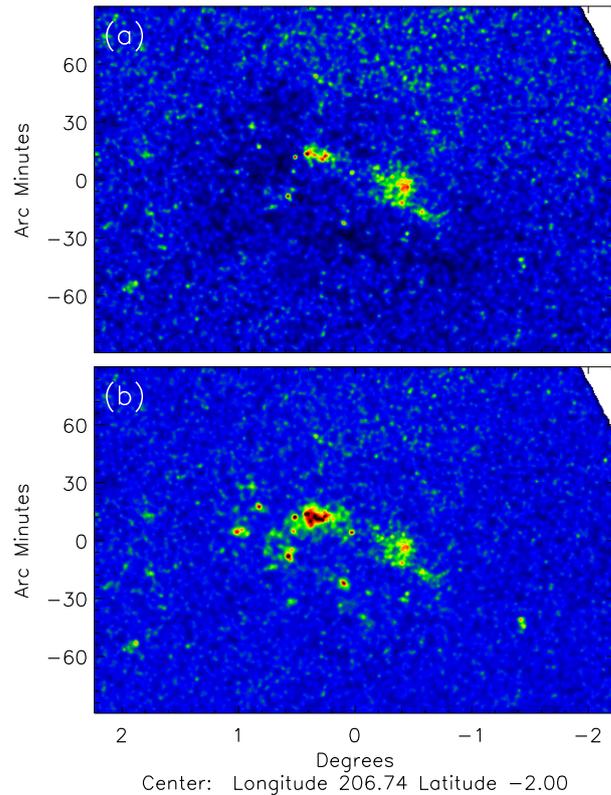


FIGURE 7.1 – Cartes de densité stellaire de la nébuleuse de la Rosette issue de [Cambrésy et al. \(2013\)](#). Le panneau du haut correspond à l’observation sans correction tandis que le panneau du bas corrige de l’extinction le long de la ligne de visée.

lisée consiste à construire une carte d’extinction prenant en compte l’obscurcissement intégré le long de la ligne de visée. La Fig. 7.1 montre un exemple de cette technique appliquée à la nébuleuse de la Rosette. Dans cet exemple, une proportion significative des étoiles en train de se former sont cachées par le nuage qui leur donne naissance. Notons que cette technique n’est pas parfaite car même les nuages en arrière-plan de la région considérée, sont intégrés à la correction.

Il serait intéressant de prendre en compte cette limitation pouvant modifier la structure apparente des modèles GDF à grande échelle. En prenant comme référence la carte de [Green et al. \(2018\)](#) qui détaille la structure 3D du MIS de la galaxie¹. Il serait alors possible de simuler l’observation de modèles GDF à différents endroits du disque galactique dans le but de quantifier les erreurs d’identification des groupes dues à la structure fragmentée du MIS.

L’introduction de gaz au sein même des modèles GDF constituerait une étape supplémentaire. Au tout premier instant de formation d’un amas, la distribution du gaz suit la distribution des étoiles. Des cartes de densité de gaz pourraient ainsi être générées et incluses à ces modèles. Dans ce cas particulier, les surdensités seront plus obscurcies que le reste de la région ce qui pourrait contrebalancer la luminosité plus importante des étoiles massives se trouvant au cœur de ces surdensités. Dans une perspective plus dynamique, le couplage

1. <http://argonaut.skymaps.info>

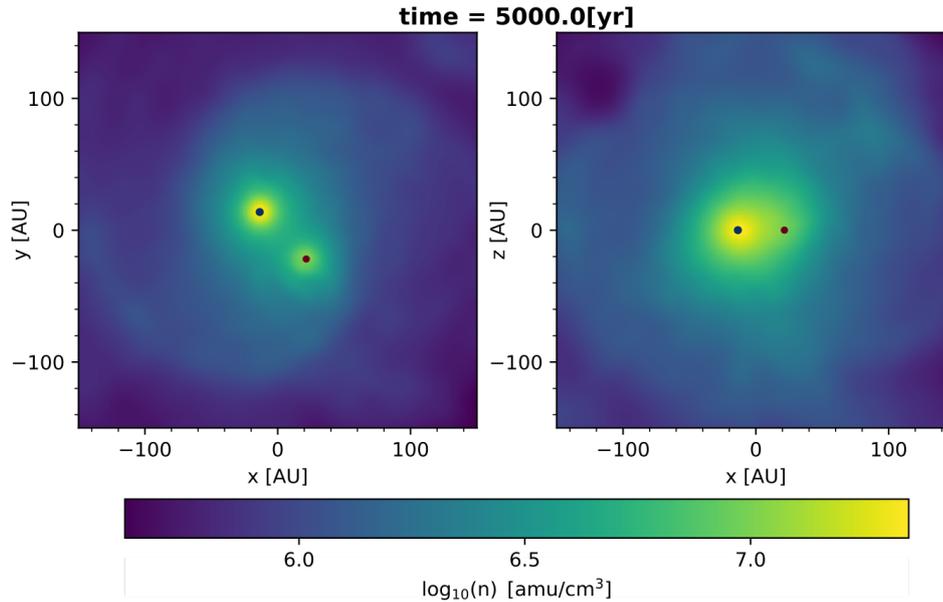


FIGURE 7.2 – Cliché à $t = 5000$ ans de l'évolution d'une binaire au sein d'un nuage de gaz correspondant initialement à un profil de Plummer. Le panneau de gauche montre la binaire ainsi que la densité du gaz dans le plan de l'orbite. Le panneau de droite correspond à un plan orthogonal. La primaire peut être identifiée par la concentration plus importante de gaz qu'elle crée. Notons qu'il n'y a pas de phénomènes d'accrétion pris en compte.

avec les effets hydrodynamiques peut être envisager grâce à la plateforme AMUSE. Dans cette veine, on pourrait s'appuyer sur les récents travaux de [Wall et al. \(2020\)](#). Ces derniers sont allés plus loin encore, en prenant en compte la rétroaction du rayonnement des étoiles sur le gaz illustrant à nouveau les nombreuses possibilités qu'offre AMUSE. Dans cette veine, nous avons exploré les effets dynamiques induit par un environnement gazeux sur l'évolution d'une binaire. Nous abordons dans la section suivante les résultats préliminaires d'un tel calcul.

7.2 Intégration d'un environnement gazeux au sein d'une binaire

Nous avons discuté de l'importance des binaires dans l'évolution dynamique des amas jeunes au chapitre 6. Outre les interactions à N -corps pouvant modifier la population des binaires ou leurs paramètres orbitaux, l'environnement gazeux présent dès leur formation implique une dissipation énergétique ayant un effet non négligeable sur leurs propriétés (voir p. ex. [Bate, 2012](#)). Dans le cadre d'un stage en lien avec cette thèse, il a été réalisé l'évolution d'une binaire immergée dans un nuage de gaz suivant un profil théorique de Plummer avec l'aide de la plateforme AMUSE. La Fig. 7.2 montre un cliché de l'évolution d'une binaire ayant un rapport de masse $q = 0.6$ après 5000 ans soit environ 1000 révolutions. On observe que de l'énergie cinétique du mouvement orbitale de la binaire est diffusée par le gaz via des ondes se propageant en spirale (panneau de gauche). On observe aussi une diffusion plus complexe du phénomène dans le plan perpendiculaire à l'orbite (panneau de droite). Cela conduit à un resserrement de l'orbite d'environ 1 U.A après les 5000 ans d'évolution

soit 2 %. Ce resserrement des binaires dû à la friction dynamique du gaz a déjà été étudié de manière analytique (Ostriker, 1999; Stahler, 2010). La prise en compte d'un environnement 3D représentatif des régions de formation d'étoiles peut constituer un sujet d'investigation futur prometteur (voir aussi Komtreff et al., 2012).

7.3 Contexte galactique

Jusqu'à présent, nous nous sommes concentrés sur les phénomènes à petite échelle œuvrant aux tous premiers stades de la vie des étoiles. Les modèles GDF ainsi construits ne s'étendant guère au delà de 10 parsecs. Néanmoins, les régions de formation d'étoiles forment un continuum de structures s'étendant sur des centaines de parsecs (Gouliermis, 2018). Il semble donc important d'envisager leur évolution dans un contexte à plus grande échelle.

L'approche aux N -corps des modèles GDF a l'avantage d'offrir la possibilité de les mettre aisément à une échelle plus importante. Nous pouvons donc reprendre un modèle GDF de 15 000 étoiles et lui donner un rayon initial de 10 pc. L'évolution d'un tel modèle sera strictement similaire aux simulations précédentes en unités N -corps. Néanmoins, puisque la densité a été diminuée, la phase d'expansion et donc le développement de la fragmentation, rapportée en unités SI, se fera sur une échelle de temps plus longue. La densité initiale d'un tel modèle est de l'ordre de $1 M_{\odot}/\text{pc}^3$ ce qui correspond à une phase d'expansion de 17 M.a. contrairement au 0.5 M.a. considérées jusqu'à présent. À ces échelles de temps et de distance, il semble indispensable de prendre en compte les effets environnementaux comme l'influence du potentiel galactique.

Grâce aux fonctionnalités d'AMUSE, nous avons pu explorer l'évolution d'un tel système au sein du potentiel analytique de Bovy (2015) qui est une approximation axisymétrique du potentiel de la Voie Lactée. Le modèle est situé aux coordonnées $[R, z] = [8 ; 0]$ kpc du centre galactique avec une vitesse initiale égale à celle d'une orbite circulaire. À ces coordonnées le potentiel est dominé par la composante du disque qui est modélisé par un potentiel de Miyamoto-Nagai (Miyamoto & Nagai, 1975) :

$$\Phi(R, z) = -\frac{GM}{\sqrt{R^2 + (a + \sqrt{z^2 + b^2})^2}}, \quad (7.1)$$

où M est la masse du disque tandis que a et b sont respectivement le rayon et la hauteur caractéristiques du disque. Le rapport b/a indique l'aplatissement du disque, une valeur de ~ 0.1 correspondant à celui de la Voie Lactée.

La Fig. 7.3, montre le cliché après 17 M.a., correspondant à la fin théorique de l'expansion. On remarque que cette fin n'est effective que dans le plan du disque (XY). Selon l'axe Z , l'expansion a été stoppée prématurément conduisant les structures issues de la fragmentation à fusionner plus tôt. En effet, les champs de marées dans cette direction sont de nature compressive contrairement à ceux agissant dans la direction du centre galactique qui sont de caractère expansif (Renaud et al., 2011). Cela conduit ainsi à briser la symétrie sphérique de l'amas. La relaxation de ce modèle est ainsi bien différente de celle des simulations étudiées jusqu'ici. De plus, on observe rapidement l'émergence de queues de marées laissant présager une nouvelle structure plus diluée à grande échelle. La Fig. 7.4 montre cette évolution du même modèle après 150 M.a. soit 10 temps dynamiques (et environ une rotation galactique). L'amas s'est relaxé et a rempli son lobe de Roche. Une mesure de l'extension

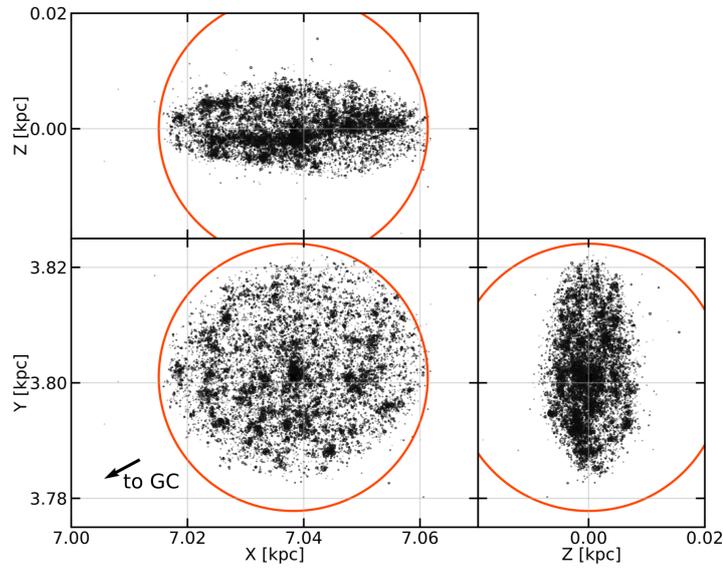


FIGURE 7.3 – Évolution d'un modèle GDF de 15 000 étoiles et de rayon initial 10 pc placé au rayon solaire d'un potentiel galactique représentatif de la Voie Lactée. Le cercle rouge correspond au rayon de marée de l'amas. La direction du centre galactique est indiquée par la flèche. Le cliché correspond à une évolution de ~ 17 M.a.. Dans ce même temps, le même modèle sans potentiel extérieur aurait conservé sa forme sphérique pour atteindre 20 parsecs de rayon dans chacune des directions.

du lobe de Roche peut être calculée par le rayon de marée r_j d'un tel modèle sous ces conditions (Binney & Tremaine, 2008) :

$$r_j = \left(\frac{GM}{4\Omega_0 A_0} \right)^{\frac{1}{3}} = 23 \text{ pc}, \quad (7.2)$$

où $M \approx 4500 M_\odot$ est la masse du système, Ω_0 et A_0 sont respectivement la vitesse angulaire et la constante de Oort correspondant au voisinage solaire.

Cette simulation est un premier test visant à inclure un potentiel perturbateur en utilisant AMUSE. Il serait intéressant de modifier les conditions initiales des modèles GDF pour rendre l'expansion initiale non-sphérique et donc moins artificielle, notamment en modifiant le champ des vitesses initiales. Elle illustre néanmoins l'importance des effets du disque galactique sur la structure des régions de formation d'étoiles à des échelles de l'ordre de seulement 10 pc.

De plus à l'ère de *Gaia*, identifier les queues de marées d'amas ouverts devient possible. Ceci a été réalisé p. ex. pour l'amas des Hyades (Röser et al., 2019; Meingast & Alves, 2019). Bien que bien plus âgés que les amas étudiés jusqu'ici (> 100 M.a.), leur caractérisation pourrait constituer un outil important pour évaluer l'efficacité de formation d'étoiles à laquelle ils ont été créés (Dinnbier & Walch, 2020).

Une approche complémentaire consiste à évaluer l'évolution du gaz du MIS aux échelles galactiques. Elle est actuellement explorée par Rieder et al. (en prép. ²). Les conditions initiales sont ici fixées par une simulation hydrodynamique de Dobbs & Pringle (2013). Les *sink particles* correspondent ici à des amas entiers qui

2. voir sa présentation <https://stevenrieder.com/presentations/chaica2/>

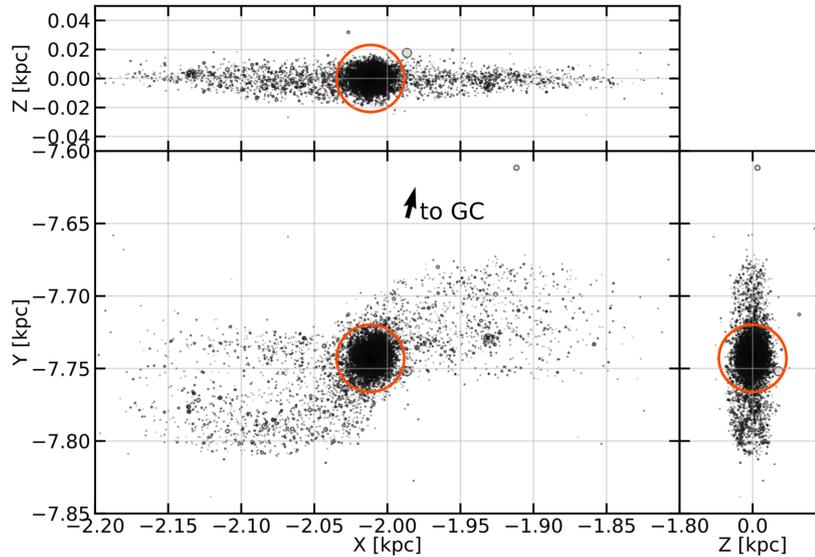


FIGURE 7.4 – Évolution du même modèle que la Fig.7.3 après 150 M.a. soit à peu près une révolution galactique.

se forment le long des bras spiraux déjà en place. Lier ces différentes échelles ainsi que la dynamique du gaz et des étoiles sur ces échelles de temps et de distance est une voie prometteuse encore très peu explorée dans les simulations de formation d'étoiles (on peut citer aussi [Tress et al., 2020](#); [Sormani et al., 2020](#), dont les travaux portent sur la formation d'étoiles dans les conditions extrêmes du centre galactique, ou CMZ pour *Central Molecular Zone*).

Cette démarche est d'autant plus importante que les simulations d'amas jeunes ne considérant pas l'environnement à plus grande échelle semblent arriver à leurs limites dans la description des processus de formations d'étoiles, notamment concernant les étoiles et régions les plus massives (voir p. ex. la revue récente de [Krumholz et al., 2019](#)). En effet, l'observation du complexe d'Orion pour ne citer que le plus étudié, montre que les étoiles s'y forment sur des périodes relativement étendues de plusieurs temps dynamiques, ce que les simulations d'un nuage seul, peinent à reproduire. L'inclusion de ces phénomènes galactiques semble ainsi cruciale dans le but par exemple de permettre le renouvellement du réservoir de gaz permettant de former de nouvelles étoiles. Cela permettra d'envisager de manière plus globale et peut-être unifiée le processus de formation d'étoiles dans notre galaxie.

BIBLIOGRAPHIE

- Bate, M. R. 2012, *MNRAS*, 419, 3115
- Binney, J. & Tremaine, S. 2008, Galactic Dynamics : Second Edition (Princeton Series in Astrophysics)
- Bovy, J. 2015, *ApJS*, 216, 29
- Cambrésy, L., Marton, G., Feher, O., Tóth, L. V., & Schneider, N. 2013, *A&A*, 557, A29
- Dinnbier, F. & Walch, S. 2020, *MNRAS*, 499, 748
- Dobbs, C. L. & Pringle, J. E. 2013, *MNRAS*, 432, 653
- Gouliermis, D. A. 2018, *PASP*, 130, 072001
- Green, G. M., Schlafly, E. F., Finkbeiner, D., et al. 2018, *MNRAS*, 478, 651
- Korntreff, C., Kaczmarek, T., & Pfalzner, S. 2012, *A&A*, 543, A126
- Krumholz, M. R., McKee, C. F., & Bland -Hawthorn, J. 2019, *ARA&A*, 57, 227
- Meingast, S. & Alves, J. 2019, *A&A*, 621, L3
- Miyamoto, M. & Nagai, R. 1975, *PASJ*, 27, 533
- Ostriker, E. C. 1999, *ApJ*, 513, 252
- Renaud, F., Gieles, M., & Boily, C. M. 2011, *MNRAS*, 418, 759
- Röser, S., Schilbach, E., & Goldman, B. 2019, *A&A*, 621, L2
- Sormani, M. C., Tress, R. G., Glover, S. C. O., et al. 2020, *MNRAS*, 497, 5024
- Stahler, S. W. 2010, *MNRAS*, 402, 1758
- Tress, R. G., Sormani, M. C., Glover, S. C. O., et al. 2020, *MNRAS*, 499, 4455
- Wall, J. E., Mac Low, M.-M., McMillan, S. L. W., et al. 2020, *arXiv e-prints*, arXiv :2003.09011

EXAMPLES OF AMUSE SCRIPTS

We describe in this annex some lines of codes to explain, with some details, the way AMUSE works (we switched to english, to spread more widely our experience with the platform).

First, we will explicit the general way to set up an N -body simulation, but it is similar for any AMUSE simulation. We will describe in particular the example script that was used to test the precision of the N -body codes on chapter 2.3.2). We will also add the method to make a head-on encounter between two stars at the center of a cluster, to show more complex manipulation with AMUSE. Then, we will describe two example scripts that allow to set GDF models : one taking advantage of the `Multiple` module and the other without.

Other examples can be found on my [Gitlab repository](#)¹ and of course on the [official AMUSE repository](#).²

A.1 Simple setup of an N -body simulation

We start by describing the basic structure of an AMUSE script. We take the example of the first chapter, section 2.3.2, testing the precision of different N -body code, on a plummer sphere in equilibrium and containing 1000 mass elements.

A.1.1 Set the initial conditions

```
In [8]: from amuse.ic.plummer import new_plummer_model
In [9]: N=1000
In [10]: stars = new_plummer_model(1000)
```

`stars` is a python object, specific to AMUSE and called a "particle set". It gathers all the information about the particles of the system we just created. It is like a table with the different particles being the rows and their attributes being the columns.

```
In [11]: stars
Out [11]: <amuse.datamodel.particles.Particles at 0x7f72b94760a0>
In [12]: print(stars)
Out [12]:
```

	key	mass	radius	x	y	z	vx	vy	vz
	-	mass	length	length	length	length	length/time	length/time	length/time
13593262122530608750	0.001	0.	0.536	-0.024	-0.003	1.155	-0.177	0.125	
453850965235164268	0.001	0.	0.859	0.364	0.962	0.656	0.144	0.491	
9049866169888506120	0.001	0.	0.390	0.125	0.579	0.052	-0.315	-0.204	
420964685199832469	0.001	0.	0.824	-0.56	0.227	-0.564	0.363	-0.120	
...	

1. https://gitlab.com/Timotchouk/tuto_amuse

2. <https://github.com/amusecode/amuse/tree/main/examples>

Each particle has a unique identifier being the key. But in practice, the user just access each element as a PYTHON list,

```
In [13]: print(stars[0])
Out [13]: Particle(13593262122530608750, set=<140130711462048>
, mass=0.001000[mass]
, radius=0.000[length]
, vx=1.155[length / time]
, vy=-0.177[length / time]
, vz=0.125[length / time]
, x=0.536[length]
, y=-0.024[length]
10 , z=-0.003[length])
```

And any attribute can be access like any object attribute in PYTHON. Here are the position in the x-direction of all particles :

```
In [14]: print(stars.x)
Out [14]: [0.536, 0.859, 0.390, 0.824, -2.026, 0.156, -0.230, -0.886, ...][length]
```

It is also possible to add any custom attribute to a particle set, just by a simple assignation :

```
In [15]: stars.id = np.arange(1000)
Out [15]: print(stars)
```

	key	id	mass	radius	x	y	z	vx	vy	vz
	-	none	mass	length	length	length	length	length/time	length/time	length/time
13593262122530608750		0	0.001	0.	0.536	-0.024	-0.003	1.155	-0.177	0.125
453850965235164268		1	0.001	0.	0.859	0.364	0.962	0.656	0.144	0.491
9049866169888506120		2	0.001	0.	0.390	0.125	0.579	0.052	-0.315	-0.204
420964685199832469		3	0.001	0.	0.824	-0.56	0.227	-0.564	0.363	-0.120
...

Here we add the attribute id to each particles of the set. It can be used for instance to identify them (instead of the key, which cannot be modified).

We can note that particle sets naturally use the ability of AMUSE to process physical quantities to ease computation for the user. In the above example, the Plummer system is initialised in computation unit by default. To give a physical meaning to the cluster we have to create a converter. It uses the almost universal convention of Hénon units. To set a system of unit, we need to set 2 out of 3 basic quantities among mass, length and time. Here we choose 1000 M_{\odot} as the total mass of the system (which equals 1 mass unit) and 1 pc as 1 length units (which corresponds to the virial radius of the system).

```
In [16]: from amuse.units import units, nbody_system
In [17]: converter = nbody_system.nbody_to_si(1000 | units.MSun, 1|units.pc)
```

With this converter, it is possible to convert the former plummer sphere into a new particle set with this system of unit :

```
In [18]: from amuse.datamodel import ParticlesWithUnitsConverted
In [19]: stars_si= ParticlesWithUnitsConverted(stars,
converter.as_converter_from_si_to_nbody())
```

Or just recreate one from scratch :

```
In [20]: stars_si = Plummer(1000, convert_nbody=converter)
In [21]: print(stars_si)
Out [21]:
```

key	mass	radius	vx	vy	vz	x	y	z
-	kg	m	$m * s^{-1}$	$m * s^{-1}$	$m * s^{-1}$	m	m	m
7928771128110001517	1.989e+30	0.000e+00	-1.332e+03	-7.460e+02	-8.358e+02	-7.781e+15	-1.166e+16	-9.442e+15
9005522656639004603	1.989e+30	0.000e+00	4.845e+02	1.358e+03	9.395e+01	-4.528e+16	2.469e+14	-2.537e+16
4042442263229067172	1.989e+30	0.000e+00	8.126e+02	7.611e+02	5.284e+02	5.281e+14	-4.957e+15	-9.928e+15
10074456913342131418	1.989e+30	0.000e+00	8.579e+02	-4.849e+02	-1.205e+03	1.221e+16	-8.339e+14	-8.235e+15
...

Particle sets are one of the great strength of AMUSE since it allows to gather a whole configuration of particles into a single PYTHON object. They allow lots of manipulations of the particles, as simple as NUMPY arrays. For instance to select all the particles which position along the x-axis is positive :

```
In [22]: selection = stars[stars.x > 0 | units.pc]
In [23]: len(selection)
Out [23]: 533
```

selection represents only a subset view on the original stars particle set, i.e. the particles in selection are the same than in stars, any modification will affect both :

```
In [24]: selection
Out [24]: <amuse.datamodel.particles.ParticlesSubset at 0x7f98c365b730>
```

In order to create a distinct particle set, you have to use :

```
In [25]: selection_distinct = selection.copy()
Out [25]: <amuse.datamodel.particles.Particles at 0x7f98d4996d60>
```

There are also plenty handful shortcuts to make these manipulation even more easy. For instance, with the attribute position, we access to the 3-dimensionnal position vector. The method lengths() enable to compute the norm of a 3-dimensionnal vector. Therefore to select the particles centered within 0.1 pc of the system :

```
In [26]: center = stars[stars.position.lengths() < 0.1 | units.pc]
In [27]: from matplotlib import pyplot
In [28]: pyplot.scatter(stars.x.value_in(units.pc), stars.y.value_in(units.pc),
                        c='green')
In [29]: pyplot.scatter(center.x.value_in(units.pc), center.y.value_in(units.pc),
                        c='red')
In [30]: pyplot.show()
```

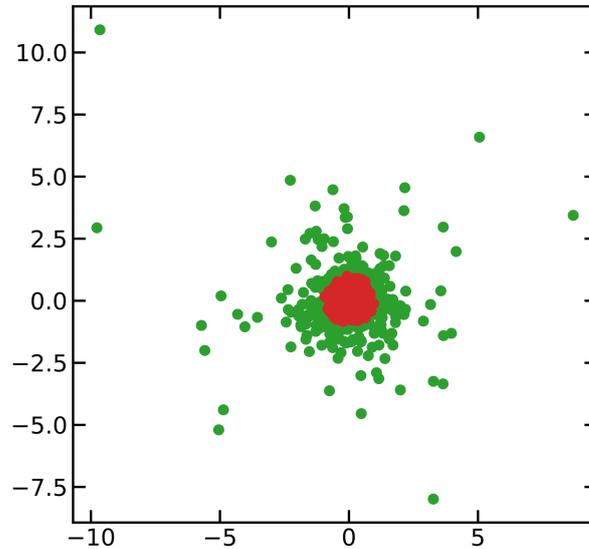


FIGURE A.1 – Selection (red dots) of the particles within a radius of 1 pc of a Plummer sphere (green dots).

All the methods to set up initial conditions and manipulate the Particle sets can be found on the folder `amuse/src/amuse/datamodel`. Particle sets are at the heart of how AMUSE works and they are very efficient to manipulate. Moreover, it exists a lots of methods that can be directly applied to a particle set and give access to basic quantities such as energies, angular momentum, radii, central density, etc.. There is no comprehensive list but they can directly be found in `amuse/datamodel/particles_attributes.py`. Either it allows to directly see what computation is done.

A.1.2 Set the integrator

Once we have initialised the initial conditions, we have to choose the integrator that will make the computation. Again, the different codes in AMUSE looks like PYTHON object but act as modules. When invoked, they launched a subprocess independant of the PYTHON script that will make the computation. The initialisation is simple :

```
In [31]: from amuse.community.ph4.interface import ph4
In [32]: gravity = ph4(mode='cpu', number_of_workers=4, redirection='file',
                    redirect_file='ph4.log')
In [33]: print(gravity)
Out [33]: <amuse.community.ph4.interface.ph4 object at 0x7f6dbbeb37f0>
```

Here we initialise the code `ph4`, that will work on a 4 cpus and which standard output will be written in the file "ph4.log". We can then configure the parameters of the code as we wish, for instance to adjust the smoothing length of the N -body integrator :

```
In [34]: grav.parameters.epsilon_squared = 0 | nbody_system.length**2
```

The default parameters can be use for a first rough computation. However to fine tune the computation, one must dig into the `interface.py` script of the code (here in `amuse/src/amuse/community/ph4`) that

make the communication between the platform and the source code, to see the relevant parameter it is using. If needed, one can also have a look to the source code. As a general rule it is always safe to make some tests.

Now, the code need the initial conditions we set up before :

```
In [35]: gravity.particles.add_particles(stars)
```

The code gravity has got his own particle set on which the computation will be done. That s what we do in the next section.

A.1.3 Make the computation and store the results

To start the computation we just have to specify the time at which the computation will stop :

```
In [36]: time = 1|unist.Myr
In [37]: grav.evolve_model(time)
```

This last calling sequence acts on the process that ran in backup since its initialisation and which makes the computation. Depending on the specified time, it may take more or less CPU time. When finished, the process return to an idle state and wait for another instruction.

We mention again that the code does not evolve the stars particle set itself, but a copy of it that we put within the integrator at In[28], which is `gravity.particles`. This particle set is similar to the one we already defined (such as `stars`) but it is in read-only mode. Therefore AMUSE provide a method to update the properties between particle sets. This method works only in one direction, i.e. we initialise a channel to update the attributes from the N -body solver to the stars particle set :

```
In [38]: channel = gravity.particles.new_channel_to(stars)
```

Then to achieve the update :

```
In [39]: channel.copy()
```

Now the stars particle set has been evolved and we can save this new configuration :

```
In [40]: from amuse.io import write_set_to_file
In [41]: write_set_to_file(stars, '/home/user/data/filename', format='csv')
```

Different format can be used : CSV, HDF5 and even a dedicated AMUSE format also exists (a binary format based on HDF5). Also, to load a specific particle set :

```
In [42]: from amuse.io import read_set_from_file
In [43]: stars_loaded = read_set_from_file('/home/user/data/filename',
                                         format='csv')
```

A.1.4 End the script

At the end of the script, one must not forget to stop the idling process of the code to shut it down correctly :

```
In [44]: gravity.stop()
```

A.1.5 Summary

These different steps describe the general procedure to follow to build an AMUSE script. However, in a simulation, we usually want to make several snapshot of the computation. Therefore we generally repeat the evolve sequence in a loop and potentially make some intermediate computations. For instance, the following code snippet sums up all the procedure. It was used to test the precision of the N -body codes in chapter 2.3.2. Note that we test the code one after each other just by using them as classical PYTHON methods.

```
#!/usr/bin/python3.8
# coding: utf8
"""
Script to test the precision of different N-body codes on a Plummer sphere during
1 nbody time
"""
# import some general libraries
import numpy as np
import time

10 # import some AMUSE methods
from amuse.units import nbody_system
from amuse.ic.plummer import new_plummer_model

# import the codes we want to test
from amuse.community.ph4.interface import ph4
from amuse.community.huayno.interface import Huayno
from amuse.community.hermite.interface import Hermite
from amuse.community.phigrape.interface import PhiGRAPE
from amuse.community.sakura.interface import Sakura

20
# Generate the initial conditions
Nstars = 1000
plummer_ini = new_plummer_model(Nstars)
plummer_ini.scale_to_standard()# correct the velocity of the stars to set the
plummer sphere in equilibrium
plummer_ini.move_to_center()# move the center of mass of the system at
coordinates (0; 0; 0)

# we can manipulate AMUSE codes as method. For instance, we will test them one
after each other in a loop.
codes = [ph4, Hermite, PhiGRAPE, Huayno, Sakura]

30 # and save their performance in dictionaries
Edrift = {'ph4':[],
          'ph4_GPU':[],
          'Hermite':[],
          'PhiGRAPE':[],
          'Huayno':[],
          'Sakura':[]}
```

```

computation_time = {'ph4':[],
                    'ph4_GPU':[],
40    'Hermite':[],
                    'PhiGRAPE':[],
                    'Huayno':[],
                    'Sakura':[]}

# We evolve the system during 10 nbody times and evaluate the drift in accuracy
# every 1 nbody time. We will estimate the mean at the end.
dt = 1. | nbody_system.time
t_end = 10. | nbody_system.time

# for each code we repeat the same loop
50 for code in codes:

    # We initialise the code
    gravity = code(number_of_workers=4)
    codename = gravity.model_name

    # Here we specify the precision of the integrator. We could have put it in
    # one more loop to test different precision factors. It is linked to the choice
    # of the timestep for the integration and is code dependant. That is why there
    # are different parameter name.
    precision_param = 0.1 #[0.32, 0.16, 0.08, 0.04, 0.02, 0.01, 0.005, 0.001]
    if codename is 'Hermite':
        gravity.parameters.dt_param = precision
60    elif codename is 'Sakura':
        gravity.parameters.timestep = precision/10 | nbody_system.time
    else:
        gravity.parameters.timestep_parameter = precision

    # As the plummer system will be evolved, we make a copy of the original
    # particle set to be able to repeat the loop with a different code
    stars = plummer_ini.copy()

    # Adding the system to the gravity solver
    gravity.particles.add_particles(stars)
70

    # We create a channel from gravity solver to the stars particle set in order
    # to be able to update the phase space configuration of the system.
    channel = gravity.particles.new_channel_to(stars)

    # We compute the initial total energy of the system to check the integration
    # precision
    E = [stars.kinetic_energy() + stars.potential_energy(G=nbody_system.G)]

```

```

# initial time of the system
t = 0 | nbody_system.time
80 t0 = time.time() # store the cpu clock time

# we evolve the system of 10*1 nbody time, so we make a loop to save energy
at each step
while t < t_end:

    t += dt
    # we integrate the system until time "t". Pay attention, we specify an
    absolut time, not a timestep!
    gravity.evolve_model(t)

    # We update the phase space configuration of the system
90 channel.copy()

    # We compute the new total energy of the system
    E.append(stars.kinetic_energy() + stars.potential_energy(G=nbody_system.G
))

# Compute the cpu time
comptime = (time.time()-t0) / 10
computation_time[codename].append(comptime)
print(codename, comptime)

100 # Compute the mean energy difference between 2 timesteps
dE = np.mean(np.abs(np.diff(E))/E[:-1])
Edrift[codename].append(dE)

# Do not forget to stop the gravity solver in order to stop the worker that
work outside of the python script!
gravity.stop()

```

A.2 Further manipulation of an AMUSE particle set

We add in this section an example to show the strenght of AMUSE particle sets. This is a function which was used in chapter 2.3.4 to make an head-on collision between two stars at the center of a plummer cluster without modifyng its equilibrium state. Indeed, when building such a cluster the direction of the velocity vectors are chosen randomly. With this function we align the direction of the velocity vectors of the two most centered stars, in order to produce a high energy encounter.

```

def make_headon_coll(stars):
    # 1. select the 2 stars closest to the cluster centre.
    # move the center of mass of the system to the origin
    stars.move_to_center()

```

```

# add an attribute "distance" to the stars particle set; it is simply the
norm of the position vector, i.e the spherical radius
stars.distance = stars.position.lengths()

# Sort the particle set according to this new attribute and select the 2
first particles (the closest to the center) in a new particle subset
10 center = stars.sorted_by_attribute('distance')[:2]

# 2. Change the velocity direction.
# compute the direction vector between the 2 particles
direc = center.position[1] - center.position[0]

# loop over both stars to recompute their velocity vectors to align them,
while conserving their norm
for part in center:
    ratio = np.sqrt(part.velocity.lengths_squared()/direc.lengths_squared())
    part.velocity = direc * ratio
20

    # take the opposite of the direction vector to have velocity of the next
particle in the loop (center[1]) in opposite direction
    direc = -direc

# 3. Return the results.
# as the "center" particle set is a subset view of the initial "stars"
particle set, the latter has got the same modifications. So we don't have to
worry about the update.
return stars

```

A.3 Example script to build a GDF model

We expose in this section two scripts that was used to produce GDF models similar to those studied through this thesis. The first uses only the direct N -body code `ph4` and the second uses it coupled with the `Multiple` module to refine the precision of the computation concerning binaries and multiple system.

A.3.1 Using only a direct N -body model

```

#!/usr/bin/python3.8
# coding: utf8
"""
Build a GDF model model with the multiple module
"""
import pickle
import numpy as np
import matplotlib.pyplot as plt
import time as check

```

10

```

from amuse.community.ph4.interface import ph4
from amuse.datamodel import Particles, ParticlesWithUnitsConverted
from amuse.io import write_set_to_file
import amuse.units.nbody_system as nbo
import amuse.units.units as u
from amuse.support.console import set_printing_strategy
set_printing_strategy("custom", preferred_units=[u.MSun, u.parsec, u.Myr],
                    precision=9, prefix="", separator="[" , suffix="]")

20 def plot_XY(stars, t_si, R_lim):
    """
    Plot on a matplotlib figure the xy-coordinate of a particle set
    :param stars: particle set to plot
    :param t_si: time label
    :param R_lim: spatial limits of the figure
    :return: Figure and axe
    """
    fig, ax = plt.subplots()
    ax.scatter(stars.x.value_in(u.pc),
30         stars.y.value_in(u.pc),
            s=stars.mass.value_in(u.MSun)*10,
            edgecolors='k', facecolor=(0.8, 0.8, 0.8, 0.5))
    ax.set_xlim([-R_lim, R_lim])
    ax.set_ylim([-R_lim, R_lim])
    ax.set_xlabel('X [pc]')
    ax.set_ylabel('Y [pc]')
    ax.set_title("t = {:.2f}".format(t_si.value_in(u.Myr)))
    return fig, ax

40
def log_simple(stars, E0=None):
    """
    Print energy log and compute energy errors
    :param stars: particle set to compute energy on
    :param converter: if stars is in SI units, to convert to nbody system
    :return: total energy, energy drift and total energy drift for next step
    """
    Ep = stars.potential_energy(G=nbo.G)
    Ek = stars.kinetic_energy()
50    Et = Ep + Ek
    Q = Ek / -Ep
    dE = (Et - E0) / (np.abs(E0.number)|nbo.energy) if E0 is not None else 0
    print(" Etot: {:.4f}, Ep: {:.4f}, Ek: {:.4f}".format(Et.value_in(nbo.energy),
                                                    Ep.value_in(nbo.energy),
                                                    Ek.value_in(nbo.energy),
                                                    ))
    print("dE: {:.3e} | Qvir: {:.4f}".format(dE, Q))
    return Et, Q, dE, Ek, Ep

```

```

def E_drift(energy):
60     """
        Plot energy drift of a simulation
        :param energy: dictionary where time, Etot and dE of a simulation are stored
                        as np.array
        :return: Figure and axe
        """
        fig, ax = plt.subplots()
        ax.plot(energy['time'], energy['dE'])
        ax.plot(energy['time'], energy['Etot'] - np.mean(energy['Etot']))
        ax.set_xlabel('time [nbody units]')
70     ax.set_ylabel('Energy variations [nbody units]')
        ax.legend(['$dE = \frac{E_i - E_{i-1}}{E_{i-1}}$',
                  '$E_{mean}$' + ' = {}'.format(np.mean(energy['Etot']))])

        return fig, ax

def maschberger_imf(N, mass_min=0.01|u.MSun, mass_max=125|u.MSun, mu=0.2|u.MSun):
    """
    Compute the IMF according to the L3 Maschberger 2013 description
    (http://adsabs.harvard.edu/abs/2013MNRAS.429.1725M)
80     :arg N: number of stars
        Truncation values for the stellar masses :
        :arg mass_max: upper bound
        :arg mass_min: lower bound
        :arg mu: mean mass of all stars
        :return: MF [u.MSun], analytical pdf, analytical cdf
    """

    # Global parameter :
    # alpha: high mass power law exponent
90     # beta: lowm ass power law exponent
    # mu: scale parameter (nearly the peak of the distribution)
    alpha = 2.3
    beta = 1.4
    mu = mu.value_in(u.MSun)

    mass_min = mass_min.value_in(u.MSun)
    mass_max = mass_max.value_in(u.MSun)

    # Define an auxiliary function G
100     G = lambda m: ( 1 + (m/mu)**(1-alpha) )**(1-beta)

    # Cumulative distribution function L3-IMF :
    cdfL3 = lambda m: ( G(m) - G(mass_min) ) / ( G(mass_max) - G(mass_min))
    # Density distribution function
    # normalisation
    A = (1-alpha)*(1-beta)/mu / ( G(mass_max) - G(mass_min) )

```

```

pdfL3 = lambda m: A*(m/mu)**(-alpha) * ( 1+ (m/mu)**(1-alpha) )**(-beta)

# The inverse of the cumulative function is known in analytic form :
# this allows to generate n masses directly from a list of random deviates
110 invPl3 = lambda m: mu * ( (G(mass_min) + (G(mass_max)-G(mass_min))*m )
** (1./(1-beta)) -1)**(1./(1-alpha))
n = np.random.random(N)
mass_distribution = invPl3(n)

return mass_distribution | u.MSun, pdfL3, cdfL3

def GDF_ic(Nstar, H0, R0=1|nbo.length):
    """
120 Create a uniformly populated sphere of stars with radial velocities

    These initial conditions provide an expansion of a star cluster leading to
    form auto-coherent fragmented configuration of stars or the so-called
    gravity-driven fragmentation (GDF) models
    :param Nstar: Number of stars in the cluster
    :param H0: expansion parameter in time^-1
    (looks like the hubble parameter in cosmology)
    :param R0: physical size of the initial sphere
    :param imf: mass functinal arguments for the MF (mass_min, mass_max...)
130 :return: initial phase spac configuration e of a Hubble-Lemaitre expansion
    in an AMUSE particle set
    """
    # Give distance to center R
    X = np.random.uniform(0, 1, Nstar)
    R = R0 * X ** (1. / 3)

    # Give random angles to stars
    X1 = np.random.uniform(0, 1, Nstar)
    X2 = np.random.uniform(-1, 1, Nstar)
140 phi = 2 * np.pi * X1
    theta = np.arccos(X2)

    # convert into cartesian coordinates
    x = R * np.cos(phi) * np.sin(theta)
    y = R * np.sin(phi) * np.sin(theta)
    z = R * np.cos(theta)

    # Build amuse particle set
    uni_sphere = Particles(Nstar)
150 uni_sphere.x = x
    uni_sphere.y = y
    uni_sphere.z = z
    uni_sphere.vx = H0 * x

```

```

uni_sphere.vy = H0 * y
uni_sphere.vz = H0 * z
return uni_sphere

def apex_GDF(H0):
160 """
    Compute theoretical timescale and radius of GDF models at the end of the
    expansion phase (in Hénon units)
    :param H0: expansion parameter in [1/nbody_system.time]
    :return: apex time and radius in Hénon units
    """
    E = 2 / H0**2
    x_apex = E/(E-1)
    theta = np.arcsin(1 / x_apex**0.5)
    t_apex = (E*(np.pi/2 - theta) + (E-1)**0.5) / H0*(E-1)**(-3/2)
170 return t_apex | nbo.time, x_apex | nbo.length

if __name__ == "__main__":

    # Initial parameters
    saving_dir = "/home/Users/data/test_GDF_model/"
    simuname = 'GDF_direct'
    fig_dir = saving_dir + "fig/"
    Nstar = 500
180 unit = nbo.length
    Mmax = 10 | u.MSun
    Mmin = 0.01 | u.MSun
    L = 0.1 | u.parsec
    H0 = 1 | nbo.time**-1
    grav_code = ph4
    mode = 'cpu'
    nbr_w = 4
    timestep_param = 0.1
    eps = 0. | nbo.length
190 snap_dt = .01 | nbo.time

    # Generate initial conditions
    uni_sphere = GDF_ic(Nstar, H0)
    t_apex, R_apex = apex_GDF(H0.number)
    # Give masses to the stars
    MF, _, _ = maschberger_imf(Nstar, mass_min=Mmin, mass_max=Mmax)
    Mtot = MF.sum()
    # Normalise star masses to be in Hénon unit
    uni_sphere.mass = (MF / Mtot) | nbo.mass
200

```

```

# Create a particle set in SI units
conv = nbo.nbody_to_si(Mtot, L)
uni_sphere_si = ParticlesWithUnitsConverted(uni_sphere,
                                             conv.as_converter_from_si_to_nbody())

# Print some usefull quantities
t = 0 | nbo.time
time = [0]
210 print("Time: ", conv.to_si(t))
print('Nstars:', len(uni_sphere), " - Mtot:", Mtot)
E0, Q, dE, ek, ep = log_simple(uni_sphere)
Etot = [E0.number]
dEtot = [dE]
Qvir = [Q]
Ek = [ek.number]
Ep = [ep.number]

# Create files to save the phase space configuration evolution during the
simulation
220 # Save initial conditions
filename = saving_dir+simuname
try:
    write_set_to_file(uni_sphere_si.savepointsavepoint(t),
                    filename,
                    'amuse', version=2.0)
    print("Amuse file created in '{}'.format(saving_dir+simuname))
except FileExistsError as e:
    raise FileExistsError("simulation results already exist, "
                          "don't overwrite results at t=0") from e
230

# Plot scatter plot of the XY-plane
plt.ioff()
R_lim = (conv.to_si(R_apex) * 1.1).value_in(u.parsec)
fig, ax = plot_XY(uni_sphere_si, conv.to_si(t), R_lim)
fig.savefig(fig_dir + "{}_{:06.3f}".format(simuname, 0).replace('.', ''))
plt.close()

# Initialise gravity solver
gravity = grav_code(mode=mode, number_of_workers=4)
240 gravity.parameters.epsilon_squared = eps ** 2
gravity.parameters.timestep_parameter = timestep_param

# Add the initial conditions within the code
gravity.particles.add_particles(uni_sphere)
# Set the channel to update data between the 2
to_stars = gravity.particles.new_channel_to(uni_sphere)

```

```

# Evolve the cluster, make several snapshots of the evolution
250 t0 = check.time()
while t < t_apex:

    t += snap_dt
    gravity.evolve_model(t)
    # Update the phase space configuration
    to_stars.copy()

    # Save the updated particle set and some usefull quantities
    try:
260     write_set_to_file(uni_sphere_si.savepointsavepoint(t),
                        filename,
                        'amuse', version=2.0,
                        append_to_file=True)
        print("Amuse file '{}' updated".format(filename))
    except OSError:
        print("AMUSE FILE '{}' NOT UPDATED".format(filename))

    print("Time: ", conv.to_si(t))
    print('Nstars: ', len(uni_sphere), "Mtot: ", Mtot)
270 time.append(t.number)
    E0, Q, dE, ek, ep = log_simple(uni_sphere, E0=E0)
    Etot.append(E0.number)
    Qvir.append(Q)
    dEtot.append(dE)
    Ek.append(ek.number)
    Ep.append(ep.number)

    # Save a scatter plot
    fig, ax = plot_XY(uni_sphere_si, conv.to_si(t), R_lim)
280 fig.savefig(fig_dir +
                "{}_{:06.3f}".format(simuname, t.number.replace('.', '')))
    plt.close()
    print('<>' * 30, '\n')

    # Compute the CPU time of the computation
    comptime = check.time() - t0
    print('COMPUTATION FINISHED: {}h {}min {:.20f}sec'
          .format(int(np.floor(comptime / 3600.)),
                  int(np.floor(comptime % 3600. / 60)),
290 comptime % 3600 % 60))

    # Shut down the subprocess of the N-body code
    gravity.stop()

```

```

# Save energy log (in Hénon unit) in a dictionary and then in a pickle file
energy = {"time": np.asarray(time),
         "Etot": np.asarray(Etot),
300         "dE": np.asarray(dEtot),
         "Ek": np.asarray(Ek),
         "Ep": np.asarray(Ep)}
with open(saving_dir+"energy_log", 'wb') as f:
    pickle.dump(energy, f)

# Plot the energy drift to control the accuracy of the computation
fig, ax = E_drift(energy)
fig.savefig(saving_dir+"E_drift")

```

A.3.2 Using the Multiples module

(Some of the methods are reused from the previous script in particular for generating the initial conditions)

```

#!/usr/bin/python3.8
# coding: utf8
"""
Build a GDF model model with the multiple module.
It processes differently single stars an multiple system which are compute in
the center of mass (CM) hypothesis. The Multiple module uses 3 codes:
- the top-level solver integrates all CM (choose a general direct N-body code)
- the small-N computes encounters between stars and create multiple system
- the kepler code computes the orbital parameters of the multiple system
10 (higher order systems are always hierarchical and can be described
    as multiple binary system).
Single stars and CM of multiple system with an extra attribute in the particle
set which is id. for multiple system id>1000000.
More infos on the class Multiples which host the multiple module under amuse/
couple/multiples
"""

import numpy as np
import matplotlib.pyplot as plt
20 import time as check
import pickle

from amuse.couple import multiples as modemulti
from amuse.community.kepler.interface import Kepler
from amuse.community.ph4.interface import ph4
from amuse.community.smalln.interface import SmallN

from amuse.io import read_set_from_file, write_set_to_file
from amuse.datamodel import Particles, ParticlesWithUnitsConverted
30 from amuse.units import nbody_system as nbo, units as u, constants

```

```

from amuse.support.console import set_printing_strategy
set_printing_strategy("custom", preferred_units=[u.MSun, u.parsec, u.Myr],
                    precision=9, prefix="", separator="[, suffix="]")

# We use a particular solver to compute encounter between particles call SmallN
# It needs these function to work properly
SMALLN = None
def init_smalln(converter):
40     global SMALLN
        SMALLN = SmallN(convert_nbody=converter)

def new_smalln():
        SMALLN.reset()
        return SMALLN

def stop_smalln():
        global SMALLN
        SMALLN.stop()
50

def newCMpos(multi_sys, CM_now):
    """
    Update the position and velocity of the center of mass of a multiple system
    to its new position.
    :param multi_sys: Multiple system in a Particle set at the time of its
        creation, first particle is the center_of mas the 2nd and 3rd, the inner
        components.
    :param CM_now: Position of the particle set at new time
    :return: Updated CM
60     """
        CM_at_encou = multi_sys[0]

        dx = CM_now.x - CM_at_encou.x
        dy = CM_now.y - CM_at_encou.y
        dz = CM_now.z - CM_at_encou.z
        dvx = CM_now.vx - CM_at_encou.vx
        dvy = CM_now.vy - CM_at_encou.vy
        dvz = CM_now.vz - CM_at_encou.vz

70     multi_sys.x += dx
        multi_sys.y += dy
        multi_sys.z += dz
        multi_sys.vx += dvx
        multi_sys.vy += dvy
        multi_sys.vz += dvz

    return CM_at_encou

```

```

def plot_XY(singles, bina, multi, t_si, R_lim):
    """
80   Plot on a matplotlib figure the xy-coordinate of a particle set
    :param singles: single particles in black
    :param bina: binary particles in red (plot only center of mass)
    :param multi: other multiple system in orange (plot only center of mass)
    :param t_si: time label
    :param R_lim: spatial limits of the figure
    :return: Figure and axe
    """

    fig, ax = plt.subplots()
    ax.scatter(singles.x.value_in(u.pc),
90         singles.y.value_in(u.pc),
            s=singles.mass.value_in(u.MSun)*10,
            edgecolors='k', facecolor=(0.8, 0.8, 0.8, 0.5))
    if not bina.is_empty():
        ax.scatter(bina.x.value_in(u.pc),
            bina.y.value_in(u.pc),
            s=bina.mass.value_in(u.MSun)*10,
            edgecolors='r', facecolor=(0.8, 0.8, 0.8, 0.5))
    if not multi.is_empty():
100        ax.scatter(multi.x.value_in(u.pc),
            multi.y.value_in(u.pc),
            s=multi.mass.value_in(u.MSun)*10,
            edgecolors='orange', facecolor=(0.8, 0.8, 0.8, 0.5))
    ax.set_xlim([-R_lim, R_lim])
    ax.set_ylim([-R_lim, R_lim])
    ax.set_xlabel('X [pc]')
    ax.set_ylabel('Y [pc]')
    ax.set_title("t = {:.2f}".format(t_si.value_in(u.Myr)))
    return fig, ax

110
def log_multi(code, conv, E0=None):
    """
    Simple diagnostics on multiples
    """

    ke = conv.to_nbody(code.kinetic_energy)
    pe = conv.to_nbody(code.potential_energy)
    Q = - ke / pe
    Nmul, Nbin, Emul = code.get_total_multiple_energy2()
120    print(' \n'*2, '<>'*40, '\n'*2)
    print('Time =', code.get_time())
    print('    top-level kinetic energy =', ke)
    print('    top-level potential energy =', pe)
    print('    total top-level energy =', ke + pe)
    print(' ', Nmul, 'multiples')
    print(' ', Nbin, 'binaries,')

```

```

print('    total multiple energy =', Emul)
E = ke + pe + conv.to_nbody(Emul)
print('    uncorrected total energy =', E)

130 # Apply known corrections.
Etid = code.multiples_external_tidal_correction \
      + code.multiples_internal_tidal_correction # tidal error
Eerr = code.multiples_integration_energy_error # integration error

Ecorr = E - conv.to_nbody(Etid + Eerr)
print('    corrected total energy =', Ecorr)
dE = (Ecorr - E0) / E0 if E0 is not None else 0
print('    relative energy error=', dE)

140 return Ecorr, Q, dE

def count_multi(code, stars, Nstar):
    """
    Count number of binaries, and higher order multiples
    plus the id of the multiples (check missing stars)
    """
    l = []
    for mul in stars[stars.id > 1000000]:
150         l.append(len(code.root_to_tree[mul].get_leafs_subset()))
    l = np.asarray(l)
    multN, count = np.unique(l, return_counts=True)
    print("binaries:", count[multN==2])
    print("triples:", count[multN == 3])
    print("quadruples:", count[multN == 4])
    for b in multN[multN > 4]:
        print("multiples of {} stars: {}".format(b, count[multN == b]))
    N = len(stars[stars.id < 1000000]) + sum(multN * count)
    print("Nstar :", N)
160    if N != Nstar:
        raise RuntimeError("Multiple module did not recover all stars, "
                           "missing stars: ", Nstar-N)
    else:
        return N

if __name__ == "__main__":

    # Initial parameters
170    saving_dir = "/home/Users/data/test_GDF_model/"
    simuname = 'GDF_multi'
    fig_dir = saving_dir + "fig/"
    Nstar = 2000

```

```

unit = nbo.length
Mmax = 10 | u.MSun
Mmin = 0.01 | u.MSun
L = 0.1 | u.parsec
H0 = 1 | nbo.time**-1
snap_dt = .01 | nbo.time
180 grav_code = ph4
mode = 'cpu'
nbr_w = 2 # We will use 2 other code within the multiple module, if only 4
# cores are available we need to reduce the number of cores used by the
# N-body solver accordingly
timestep_param = 0.1
eps = 0. | nbo.length

# Generate initial conditions
uni_sphere = GDF_ic(Nstar, H0)
190 t_apex, R_apex = apex_GDF(H0.number)
# Give masses to the stars
MF, _, _ = maschberger_imf(Nstar, mass_min=Mmin, mass_max=Mmax)
Mtot = MF.sum()
# Normalise star masses to be in Hénon unit
uni_sphere.mass = (MF / Mtot) | nbo.mass

# Multiple systems are identified by an id>1000000, so it is important to
# set this attribute
200 uni_sphere.id = np.arange(1, Nstar+1)

# the module detects collision based on a collisionnal radius.
# For simplicity, we set it proportionnal to the mass of the star
uni_sphere.radius = uni_sphere.mass.number / 2 | nbo.length

# Create a particle set in SI units
conv = nbo.nbody_to_si(Mtot, L)
CoMs = ParticlesWithUnitsConverted(uni_sphere,
                                   conv.as_converter_from_si_to_nbody()).
copy()

210 # Print some usefull quantities
t = 0 | nbo.time
time = [0]
print("Time: ", conv.to_si(t))
print('Nstars:', len(uni_sphere), " - Mtot:", Mtot)
E0, Q, dE, ek, ep = log_simple(uni_sphere)
Etot = [E0.number]
dEtot = [dE]
Qvir = [Q]
Ek = [ek.number]
220 Ep = [ep.number]

```

```

# Create files to save the phase space configuration during the simulation
# we create 3 different particle sets which will be updated accordingly.
# we split CoMs in 3 different particle sets
singles = CoMs[CoMs.id<1000000].copy()

# For the moment there are no multiple systems
bina = Particles()
multi = Particles()

230 filename_singl = saving_dir + simuname + '_singles'
filename_bina = saving_dir + simuname + '_binaries'
filename_multi = saving_dir + simuname + '_multiples'
try:
    write_set_to_file(singles.savepoint(t),
                      filename_singl,
                      'amuse', version=2.0)
    print("Amuse file created in '{}'.format(filename_singl))

    write_set_to_file(bina.savepoint(t),
                      filename_bina,
240                      'amuse', version=2.0)
    print("Amuse file created in '{}'.format(filename_bina))

    write_set_to_file(multi.savepoint(t),
                      filename_multi,
                      'amuse', version=2.0)
    print("Amuse file created in '{}'.format(filename_multi))

except FileExistsError as e:
250     raise FileExistsError("simulation results already exist, "
                             "don't overwrite results at t=0") from e

# Plot scatter plot of the XY-plane
# plt.ioff()
R_lim = (conv.to_si(R_apex) * 1.1).value_in(u.parsec)
fig, ax = plot_XY(singles, bina, multi, conv.to_si(t), R_lim)
fig.savefig(fig_dir + "{}_{:06.3f}".format(simuname, 0).replace('.', ''))
plt.close()

260 # Initialize top-level solver
gravity = ph4(convert_nbody=conv, mode=mode, number_of_workers=nbr_w)
gravity.parameters.epsilon_squared = eps ** 2
gravity.parameters.timestep_parameter = timestep_param

# Add the initial conditions within the code
gravity.particles.add_particles(CoMs)

```

```

# Enable the stopping condition when the distance between 2 particles is
# less than the sum of their radius
270 stop_cond = gravity.stopping_conditions.collision_detection
stop_cond.enable()

# Initialize the small-N integrator and the Kepler code
init_smalln(conv)
kep = Kepler(unit_converter=conv)
kep.initialize_code()

# Initialise the Multiple module
280 multicode = modemu.Multiples(gravity, new_smalln, kep, gravity_constant=
constants.G)

perturb_limit=0.02
# the multiple module checks before the potential creation of a binary (1 &
2),
# if one or more other particles (3) can perturb the encounter, based on this
criterion:
# if  $\text{perturb\_limit} * (m_1+m_2/a^3) > m_3/d^3$  (where m are the mass of the
particles,
# a the semi-major axis of the binary and d the distance from the perturber
to the binary)
# this test is done for all particles around the binary, for those which
passes
# they are considered as potential perturbers (see line 934 to 965 of couples
/multiples.py)
multicode.neighbor_perturbation_limit = perturb_limit

290 neighbor_veto=True
# Then if neighbour_veto is True, any presence of a perturber avoid
# the multiple module to create a stable multiple system, if False the module
# include the perturber to the SmallN calculation. Nevertheless this leads
# (at the time of the creation of this script ~2021) to unrealistically
# numerous multiple systems (see line 975 to 995 of couples/multiples.py)
multicode.neighbor_veto = neighbor_veto

# set the level of verbosity of the multiple module
300 multicode.global_debug = 2

# Set the channel to update data between module and the CoMs particle set
to_stars = multicode.particles.new_channel_to(CoMs)

# Evolve the cluster, make several snapshots of the evolution
t0 = check.time()
while conv.to_nbody(multicode.get_time()) < t_apex:

```

```

310 # launch evolution
    t += snap_dt
    multicode.evolve_model(conv.to_si(t))

    # Before update data we have to update the list of top level particles.
    # Indeed, if two particles are in a binary after the evolution, they are
    # now replaced by the center of mass of the new binary. This method
    # update the list of particles and accounts for the creation of new
    # particles
    multicode.particles.synchronize_to(CoMs)

320 # Now we can update the phase-space of the top level particles,
    # the previous method does not update the phase space configuration
    # just the list of particles
    to_stars.copy()

    # The module manage id of CM of multiple systems as "index_in_code"
    # We want to retrieve them in our particle set as "id"
    # id of multiple system are >1000000
    to_stars.copy_attribute("index_in_code", "id")

330 # Print hierarchical structure of multiple module
    multicode.print_trees_summary()

    # Create specific particle sets for multiple systems
    bina = Particles()
    multi = Particles()

    # recover the inner components of all multiple system created and save
    # them as attributes of the macro-particle representing their center of
    # mass the multiple module save all encounter in a dictionary
    # named "root_to_tree"
340 for binatree in multicode.root_to_tree.values():
    # recover center_of mass and inner component of a multiple system in
    # one particle set
    structure = binatree.get_tree_subset().copy()

    # update the new position of the CM (because it has moved since
    # the encounter), the root_to_tree save its position at
    # the time of the encounter
350 new_CM = newCMpos(structure, structure[0].as_particle_in_set(CoMs))

    # Add CM of multiple system according to the number of their
    # inner components
    if len(structure) == 3:
        bina.add_particle(new_CM)
    elif len(structure) > 3:
        multi.add_particle(new_CM)

```

```

# Check if multiple module recovers all the particles
n = count_multi(multicode, CoMs, Nstar)
360 # Save particle set with all memory of the structure
singles = CoMs[CoMs.id<1000000].copy()
try:
    write_set_to_file(singles.savepoint(t), filename_singl,
                    'amuse', version=2.0,
                    append_to_file=True)
    print("Amuse file created in '{}'.format(filename_singl))

    write_set_to_file(bina.savepoint(t), filename_bina,
                    'amuse', version=2.0,
370     append_to_file=True)
    print("Amuse file created in '{}'.format(filename_bina))

    write_set_to_file(multi.savepoint(t), filename_multi,
                    'amuse', version=2.0,
                    append_to_file=True)
    print("Amuse file updated in '{}'.format(filename_multi))

except OSError:
    print("AMUSE FILE '{}' NOT UPDATED")
380

# Get energy logs
E0, Q, dE = log_multi(multicode, conv, E0)
time.append(conv.to_nbody(multicode.get_time()).number)
Etot.append(E0.number)
Qvir.append(Q)
dEtot.append(dE)
print('Total mass', CoMs.mass.sum())

# Save a scatter plot
390 fig, ax = plot_XY(singles, bina, multi, conv.to_si(t), R_lim)
fig.savefig(fig_dir +
           "{}_{:06.3f}".format(simuname, t.number.replace('.', '')))
plt.close()
print('<>' * 30, '\n')

# Compute the CPU time of the computation
comptime = check.time() - t0
print('COMPUTATION FINISHED: {}h {}min {:.20f}sec'
      .format(int(np.floor(comptime / 3600.)),
400         int(np.floor(comptime % 3600. / 60)),
            comptime % 3600 % 60))
    
```

```
# Shut down the subprocess of all codes
gravity.stop()
kep.stop()
stop_smalln()
410
# Save energy log (in Hénon unit) in a dictionary and then in a pickle file
energy = {"time": np.asarray(time),
          "Etot": np.asarray(Etot),
          "dE": np.asarray(dEtot),
          "Ek": np.asarray(Ek),
          "Ep": np.asarray(Ep)}
with open(saving_dir+"energy_log", 'wb') as f:
    pickle.dump(energy, f)
420
# Plot the energy drift to control the accuracy of the computation
fig, ax = E_drift(energy)
fig.savefig(saving_dir+"E_drift")
```

Identification et analyse des structures stellaires émergent des régions de formation d'étoiles

Résumé

Les jeunes étoiles se forment au sein de structures fragmentées héritées du nuage de gaz dont elles sont issues, appelées régions de formation d'étoiles (SFR). Dans cette thèse, nous étudions les interactions N -corps, jouant un rôle clé dans leur évolution, grâce à des simulations numériques, développées avec la plateforme AMUSE.

En utilisant la méthode de fragmentation à N -corps (GDF), nous produisons des amas d'étoiles jeunes réalistes, sur lesquels nous testons deux outils, le MST et HOP, utilisés pour identifier les groupes sur-denses d'étoiles.

En particulier, nous évaluons la robustesse de la méthode MST lorsqu'elle est utilisée sur des observations, c'est-à-dire uniquement sur la distribution projetée 2D des étoiles. Nous montrons qu'une telle identification sous-estime les propriétés dynamiques, telle que la masse dynamique des groupes, par rapport à une identification complète en 3D.

Nous étudions aussi d'autres limites observationnelles, via la comparaison à un ensemble de groupes identifiés au sein de SFR, par la récente mission Gaia DR2.

Enfin, nous présentons les détails numériques employés pour décrire des effets complexes tels que celui des binaires, de l'extinction des poussières ou encore de l'environnement galactique.

Mot-clés : Formation d'étoiles, Amas d'étoiles, Dynamiques N -corps, Outils de détection de groupes, binaires.

Résumé en anglais

Young stars form within fragmented structures inherited from the gas cloud they emerged from, named star forming regions. Subsequently, dynamical processes play a key role in their evolution.

In this thesis, we study the N -body interactions between young stars thanks to numerical simulations, developed with the AMUSE platform.

Using the gravity-driven fragmentation (GDF) method, we produce realistic young star clusters, on which we test two tools, the MST and HOP, used to identify over-dense groups of stars.

In particular, we evaluate the robustness of the MST method. Such tool is usually applied only on the 2D projected distribution of stars because of observation limitations. We point that such identification underestimates the dynamical properties, such as measurements of radius, velocity dispersion and dynamical mass of the groups, compared to a full 3D-based identification.

We study other observational limitations by comparing the dynamical properties of our modelled groups to a set of observations of the recent Gaia DR2 mission.

Finally, we highlight the details of our simulation framework used to describe complex ingredients such as the effect of binaries, dust extinction or even the galactic environment.

Key words: Star Formation, Stellar clusters, N -body dynamics, Clump finding algorithm, Binaries