Polytechnique Montréal

PHS2107

Mécanique supérieure

Simulation à N corps

Étude Monte Carlo de la formation de systèmes planétaires stables

Équipe : — *Matricules :*
Félix Desrochers — 1847002
Olivier Gamache — 1854674
Gregory Giard — 1840858

Remis à
Pr. Xavier Daxhelet

4 Décembre 2017

Table des matières

1 Introduction

Au cours des dernières années, plusieurs projets d'envergure dont le satellite Kepler ont été mis en place afin de découvrir des exoplanètes ayant des caractéristiques permettant possiblement l'apparition de formes de vie. Or, dans de tels projets, une connaissance théorique approfondie des systèmes planétaires et des étapes de leur formation s'avèrent critiques. Ainsi, considérant l'intérêt que représente la caractérisation de la formation de systèmes planétaires, il pourrait être instructif d'étudier les conditions initiales étant les plus propices à la formation de systèmes planétaires stables étendus (comptant plusieurs planètes). Alors, dans le cadre de ce projet, nous avons étudié l'importance de divers paramètres (masse, vitesse, nombre de planètes) sur la formation d'un système planétaire stable à partir de diverses distributions de corps célestes initiaux.

Pour ce faire, nous avons créé des simulations numériques directes à N corps pouvant compter jusqu'à environ 500 corps et, à partir de ces simulations, avons tiré nos résultats d'une méthode Monte Carlo : pour certains paramètres initiaux (nombre de planètes, vitesse moyenne, masse moyenne, moment cinétique par rapport au centre du système, etc.) nous avons lancé plusieurs simulations ayant des configurations aléatoires (distribution des masses, des vitesses, des positions, etc.) afin de déterminer la probabilité qu'un système ayant de tels paramètres initiaux produise un système planétaire stable. Nous avons donc pris des échantillons pour certaines configurations initiales et avons déterminé à partir des résultats si une telle configuration est propice à la formation d'un système planétaire stable. Ces résultats nous ont permis donc de tirer certaines conclusions concernant l'importance relative de ces divers paramètres sur la formation d'un système stable.

Dans ce rapport, nous discuterons d'abord de l'algorithme utilisé. Nous nous attarderons ensuite à la discussion de l'efficacité de notre algorithme en le comparant avec les résultats analytiques pour un problème à deux corps et finirons par exposer et discuter des résultats obtenus concernant les paramètres favorisant l'émergence de systèmes planétaires stables.

2 Algorithme

Tout d'abord, nous nous attarderons à exposer le fonctionnement de l'algorithme implémenté dans notre programme. L'algorithme se subdivise en plusieurs étapes. En effet, la partie la plus primitive de l'algorithme s'occupe d'actualiser la position des planètes et doit aussi gérer les collisions entre ces dernières. Par la suite, le code permet aussi de créer un système avec des conditions initiales (nombre de planètes, répartition de la masse, répartition des vitesses, position des planètes, etc.). Nous pouvons ainsi lancer plusieurs simulations en faisant varier ces paramètres et implémenter une méthode Monte Carlo. Finalement, l'algorithme doit être en mesure de définir si un certain système est stable ou non et, dans le premier cas, trouver le nombre de planètes composant le système stable. Nous discuterons de chacune de ces étapes individuellement. Pour obtenir plus de détails techniques concernant le programme utilisé, voir l'annexe A.

2.1 Actualisation des positions

La partie la plus critique de l'algorithme est l'actualisation de la position des planètes. Effectivement, afin d'obtenir des résultats valables, le programme doit être en mesure de déplacer les planètes dans le temps de manière réaliste. Pour ce faire, nous utilisons les lois de Newton et l'intégration de Verlet. Dans un premier temps, les lois de Newton nous permettent de déterminer l'accélération que subit chacune des planètes à un moment précis. En effet, nous n'avons qu'à déterminer la force s'appliquant sur une planète à partir de toutes les autres planètes, tel qu'illustré à la figure 1.

Ainsi, à un moment fixe, nous pouvons évaluer les composantes de l'accélération que subit une planète donnée en évaluant la somme des forces qui s'applique sur celle-ci, soit :

$$\vec{a} = \sum_{i=1}^{n} \frac{Gm_i}{|\vec{r_i}|^3} \vec{r_i} \tag{1}$$

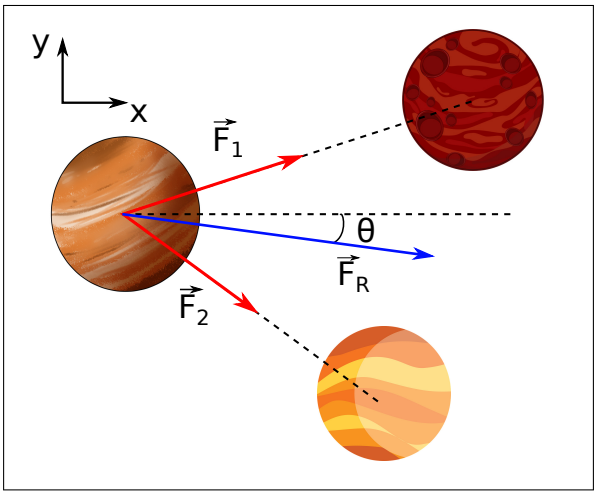

Figure 1 – Exemple de détermination de la force s'exerçant sur un corps pour un système contenant trois planètes

où n représente le nombre de planètes, m_i la masse de la i^e planète et $\vec{r_i}$ le vecteur allant de la planète sur laquelle nous calculons l'accélération vers la i^e planètes. Or, afin de pouvoir implémenter une telle formule, il est beaucoup plus pratique de décomposer cette dernière en composantes (x et y) et d'exprimer celle-ci en fonction de la position (x_i,y_i) de toutes les planètes. On obtient dans ce cas les équations suivantes :

$$a_x = \sum_{i=1}^{n} \frac{Gm_i}{(\sqrt{(x_i - x)^2 + (y_i - y)^2})^3}(x_i - x) \quad (2)$$

$$a_y = \sum_{i=1}^{n} \frac{Gm_i}{(\sqrt{(x_i - x)^2 + (y_i - y)^2})^3}(y_i - y) \quad (3)$$

où (x, y) représente les coordonnées de la position de la planète dont nous calculons l'accélération et (x_i, y_i) les coordonnées de la i^e planète. Ainsi, afin de calculer l'accélération sur chaque planète, nous n'avons qu'à calculer l'interaction entre chaque planète et en déduire par la suite l'accélération correspondante de chaque planète en parcourant chacune d'elle.

Par la suite, disposant de la position et de l'accélération des planètes à un certain moment, nous devons être en mesure de déterminer la position de chaque planète à un temps ultérieur. Afin d'accomplir une telle tâche, plusieurs méthodes numériques existent dont la méthode d'Euler ou encore Runge-Kutta. Or, l'utilisation de ces méthodes, quoique précises dans la majorité des cas, s'avère précaire dans notre cas, car plusieurs études de cas ont démontré qu'elles produisent des orbites instables [1, 3] ce qui s'avère très problématique dans une étude comme la nôtre où nous tentons justement d'étudier la stabilité de systèmes planétaires. Pour ces raisons, notre choix de méthode numérique s'est arrêté sur la méthode de Verlet puisqu'en plus d'être précise et stable, celle-ci comporte d'autres propriétés intéressantes comme la réversibilité temporelle et la préservation de la forme symplectique dans l'espace des phases, le tout, n'ayant pas une lourdeur numérique bien supérieure à la méthode d'Euler.

Pour appliquer cette méthode, nous définissons un court intervalle de temps Δt pour lequel nous actualiserons à chaque fois le système. La méthode de Verlet nécessite la position précédente et actuelle de chaque planète. Ainsi, lors de la première itération celle-ci ne peut être appliquée : nous devons actualiser la position et la vitesse de chaque planète à partir de la méthode d'Euler implicite. Donc lors de la première itération nous utilisons les équations 4 et 5.

$$v_{i+1} = v_i + a_{x,i} \cdot \Delta t \quad (4)$$

$$x_{i+1} = x_i + v_{i+1} \cdot \Delta t \quad (5)$$

,où x_i désigne la position de la planète au temps t alors que x_{i+1} désigne sa position au temps $t + \Delta t$ (de même pour sa vitesse et l'accélération). Ensuite, lors des prochaines itérations (en gardant en mémoire la position précédente), nous utilisons l'intégration de Verlet pour actualiser la position :

$$x_{i+1} = 2x_i - x_{i-1} + a_{x,i}\Delta t^2 \quad (6)$$

À partir de ces positions, nous pouvons déduire la vitesse de chaque planète :

$$v_{i+1} = \frac{x_{i+1} - x_{i-1}}{2\Delta t} \quad (7)$$

À chaque étape il est important de garder en mémoire la position précédente afin de pouvoir implémenter la méthode de Verlet. Ainsi, l'algorithme pour actualiser les positions se résume comme suit :

1. Calculer les composantes de l'accélération de chaque planète avec 2 et 3
2. Pour la première itération, actualiser la position et la vitesse à partir de la méthode d'Euler, mais en gardant en mémoire l'ancienne position
3. Calculer la nouvelle accélération
4. Actualiser les données à partir de l'intégration de Verlet
5. Recommencer les étapes 3 et 4

2.2 Collisions

Or, si nous ne faisons qu'actualiser la position des planètes sans tenir compte de leur position relative et de leur dimension respective, celles-ci peuvent se retrouver arbitrairement proches et donc subir une force arbitrairement grande. Ce phénomène éjecterait les planètes hors du système et apporterait donc des aberrations apparentes lors de la simulation. Afin d'éviter une telle situation, nous devons prendre en compte les collisions entre les planètes. Pour ce faire, nous attribuons un rayon à chaque planète en fonction de sa masse : chaque planète définit son rayon pour avoir la même masse volumique que la planète Terre. Soit,

$$r = \sqrt[3]{\frac{3m_P}{4\pi\rho_T}} \tag{8}$$

où ρ_T est la densité volumique de la Terre et m_p la masse de la planète. Par la suite, nous définissons un critère qui définit s'il y a collision entre deux planètes. Dans notre cas, nous avons défini qu'il y a collision entre deux planètes si la distance les séparant est inférieur à 95% de la somme de leur rayon. Ainsi, il y a collision si le critère suivant est respecté pour deux planètes :

$$d_{1,2} < 0.95 \cdot (r_1 + r_2) \tag{9}$$

où r_1 et r_2 sont les rayons de la première et de la deuxième planète respectivement et $d_{1,2}$ la distance les séparant.

Finalement, si nous détectons une collision nous devons gérer celle-ci. Dans notre cas, nous avons décidé de modéliser les collisions comme des collisions parfaitement inélastiques. Même si une telle approximation ne reflète pas très bien la complexité du phénomène réelle, une telle approche a déjà été utilisée dans d'autres travaux [2] et s'est avérée relativement concluante. Par conséquent, lors d'une collision, les deux planètes fusionnent pour créer une nouvelle planète dont la masse est la somme des deux précédentes, dont la position est le centre de masse des deux planètes et dont la vitesse est donnée par la conservation de la quantité de mouvement. Ce processus est illustré à la figure 2. Ainsi, les coordonnées de la nouvelle planète sont donc données par les formules suivantes :

$$\begin{cases} x_3 = \frac{m_1 x_1 + m_2 x_2}{m_1 + m_2} \\ y_3 = \frac{m_1 y_1 + m_2 y_2}{m_1 + m_2} \\ v_{3,x} = \frac{m_1 v_{1,x} + m_2 v_{2,x}}{m_1 + m_2} \\ v_{3,y} = \frac{m_1 v_{1,y} + m_2 v_{2,y}}{m_1 + m_2} \end{cases} \tag{10}$$

En résumé, afin de gérer les collisions dans notre programme, lorsque la distance séparant deux planètes est inférieure à 95% de la somme de leur rayon nous considérons que ces deux planètes subissent une collision parfaitement inélastique (fusion des planètes et conservation de la quantité de mouvement).

2.3 Prise de données par Monte Carlo

D'autre part, à partir de l'algorithme décrit ci-haut, nous devons lancer plusieurs simulations à partir de différentes distributions statistiques afin d'observer l'influence de différents paramètres sur la stabilité. Dans notre cas, les paramètres que nous avons étudiés sont la masse moyenne des objets célestes, le nombre d'objets et finalement la vitesse moyenne des planètes. Donc, pour étudier ces paramètres, nous avons effectué plusieurs simulations pour lesquels un paramètre était varié systématiquement alors que tous les autres étaient fixes. En outre, pour toutes ces simulations la répartition des positions et des vitesses était fixée aléatoirement selon une distribution statistique. Par conséquent, deux paramètres de l'algorithme sont particulièrement critiques relativement à la méthode d'échantillonnage Monte Carlo : l'initialisation des paramètres et la détermination de la stabilité d'un système donné. Nous élaborerons ces deux sujets dans les deux prochaines sections.

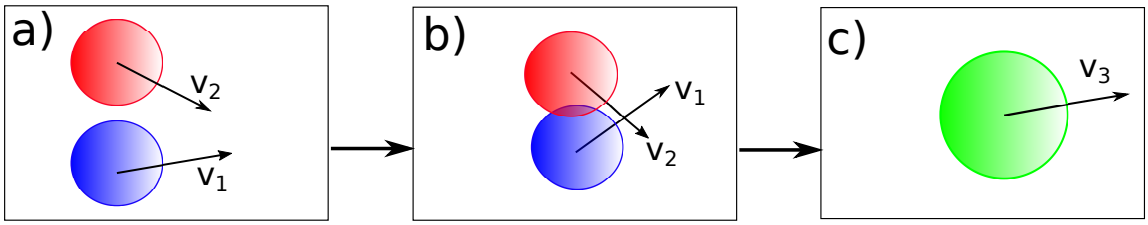

Figure 2 – Représentation de la gestion d'une collision par l'algorithme : (a) avant la collision, (b) lors de la collision, (c) nouvelle planète créée après la collision

2.3.1 Initialisation des paramètres

Dans un premier lieu, afin de pouvoir prendre des données, nous devons définir comment les différents paramètres du système sont fixés pour une simulation donnée. Effectivement, au départ de chaque simulation, nous devons initialiser le nombre de planètes, la position, la vitesse et la masse de chaque planète. Chaque paramètre est fixé selon des paramètres d'entrée et des distributions statistiques précises. Nous ferons donc le survol de la manière dont tous ces paramètres sont définis. Dans un premier temps, le nombre de planètes est fixé comme un paramètre d'entrée. Celui-ci n'est donc aucunement aléatoire. Ensuite, en ce qui a trait à la position, toutes les planètes sont définies à l'intérieur d'un cercle de rayon $r = 10^7$ m. Les positions sont donc définies en coordonnée polaire selon deux paramètres : le rayon et l'angle. Ces deux paramètres sont répartis uniformément sur leur domaine respectif. On peut donc écrire :

$$\vec{r}(r,\theta) : \begin{cases} r \sim U(0, 10^7) \text{ m} \\ \theta \sim U(0, 2\pi) \text{ rad} \end{cases} \tag{11}$$

Un exemple de configuration initiale des positions donné par le programme est illustré à la figure 3.

Dans un même ordre d'idée, la masse est fixée autour d'une masse moyenne m_μ qui est un paramètre d'entrée. En ce qui concerne sa répartition, la masse est fixée aléatoirement pour chaque planète selon une loi normale dont la variance est le tiers de la masse moyenne. On a alors pour la masse des planètes :

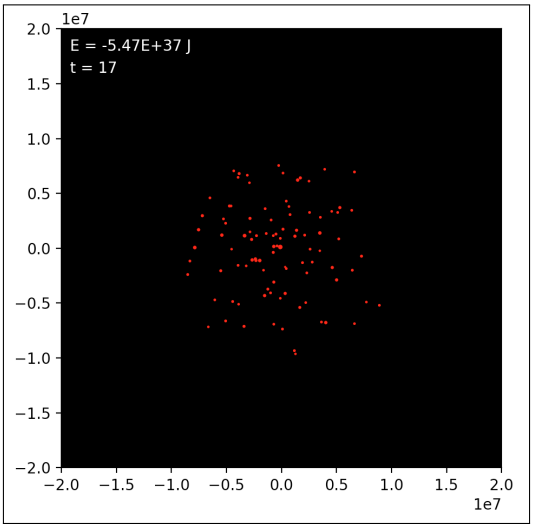

Figure 3 – Exemple de répartition des positions initiales de la simulation

$$m \sim N\left(m_\mu, \frac{m_\mu}{3}\right) \tag{12}$$

Après avoir attribué aléatoirement les masses, nous normalisons l'ensemble des masses pour que la masse moyenne soit bien m_μ. En fin de compte, concernant la vitesse, son module est fixé de manière similaire à la masse : le module moyen de la vitesse est un paramètre d'entrée v_μ et le module de chaque vitesse est fixé selon une loi normale de variance égale au tiers de la vitesse moyenne :

$$|\vec{v}| \sim N\left(v_\mu, \frac{v_\mu}{3}\right) \tag{13}$$

De même, après avoir attribué une norme de vitesse à chaque planète, celles-ci sont normalisées pour que la vitesse moyenne soit v_μ. Par la suite, la direction de la vitesse est fixée aléatoirement de

telle façon à ce que le moment angulaire moyen par rapport à l'origine soit donné par le paramètre d'entrée. De ce fait, nous avons un système complet comportant un nombre de planètes, une masse moyenne, une vitesse moyenne et un moment angulaire moyen clairement défini, mais où la position respective et l'orientation de la vitesse de chaque planètesont aléatoires.

2.3.2 Définition de la stabilité

Dans un second lieu, l'algorithme doit être en mesure de déterminer si un système est stable ou non. Pour ce faire, nous avons utilisé quelques approximations qui nous paraissent raisonnables pour définir si l'orbite d'une planète apparaît stable ou non.

Tout d'abord, avant d'essayer de définir un système stable l'algorithme attend un certain nombre d'itérations pour lequel nous avons jugé que le système devenait relativement calme, soit un temps après lequel il n'y avait plus beaucoup de collisions et après lequel la majorité des structures semblaient relativement s'être fixées. Nous avons fixé ce temps à 650 itérations. Après avoir atteint ce temps, l'algorithme trouve la planète la plus massive à l'intérieur d'une certaine région de l'espace centrée à l'origine afin de ne pas choisir une planète massive qui se serait fait éjecter du système par exemple. Une fois cette planète trouvée le programme tente de trouver le nombre de planètes en orbite autour de cette dernière. Pour ce faire, il définit une planète stable comme une planète dont la distance de la planète centrale est inférieure à un certain seuil et dont la vitesse est inférieure à la vitesse de libération, soit :

$$v_{planete} < \sqrt{\frac{2GM_{centre}}{d}} \tag{14}$$

où M_{centre} est la masse du corps centrale et d la distance séparant la planète de l'astre central. Finalement, l'algorithme suit l'évolution de ce nombre de planètes et arrête lorsque le nombre de planètes en orbite reste constant pendant plus de 100 itérations ou après avoir atteint le nombre d'itérations maximal (1000 dans notre cas). Ainsi, après avoir arrêté la simulation, l'algorithme donne le nombre de planètes en orbites, la masse de la planète centrale et le nombre de planètes restantes à la fin de la simulation.

Ainsi, à partir de l'algorithme décrit ci-haut, nous avons donc été en mesure de faire fonctionner plusieurs simulations et de juger de leur stabilité en fonction de leurs paramètres initiaux. Ces résultats, de même que l'efficacité de l'algorithme sont discutés dans les sections suivantes.

3 Résultats et discussion

3.1 Étude de l'efficacité de l'algorithme

Afin de vérifier que l'algorithme produit des résultats légitimes et réalistes, il faut comparer ces résultats à ce que la théorie donne. Bien sûr, il serait absurde de tenter de résoudre analytiquement un système à N corps. Ainsi, une simulation de deux corps en orbite stable a été effectuée. La disposition initiale est présentée dans la figure suivante :

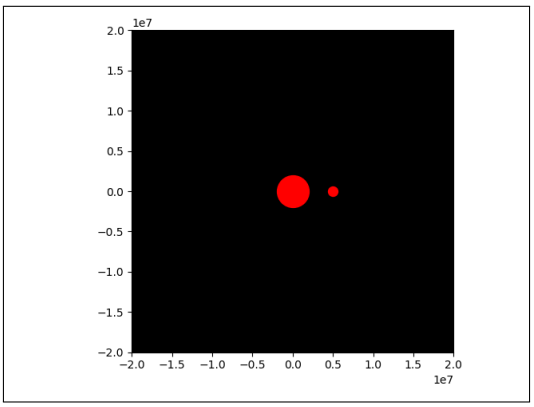

Figure 4 – Position initiale des deux masses pour la comparaison avec le modèle théorique

Ici, les conditions initiales du système ont été établies par les valeurs suivantes :

$$\begin{cases} m_1 = 1000 \cdot m_T \ \text{[kg]} \\ m_2 = m_T \ \text{[kg]} \\ r_{min} = 5000 \times 10^3 \ \text{[m]} \\ \vec{v_2} = 300000\hat{y} \ \text{[m/s]} \end{cases} \tag{15}$$

Ici, m_T est la masse de la Terre et on considère que la planète 2 est à son périapse initialement (car la vitesse est uniquement selon l'axe y à $t = 0$).

On commence donc par effectuer les calculs théoriques des différents paramètres de l'ellipse. Avec les valeurs initiales, on peut trouver la masse totale et la masse réduite du système :

$$\begin{cases} \mu = \frac{m_1 m_2}{m_1+m_2} = 5,8941 \times 10^{24} \text{ kg} \\ M = m_1 + m_2 = 5,9059 \times 10^{27} \text{ kg} \end{cases} \quad (16)$$

On peut ensuite calculer le moment cinétique du système :

$$l_z = \mu \cdot r_{min} \cdot v_2 = 8,841 \times 10^{36} \ [\text{kgm}^2/\text{s}] \quad (17)$$

On peut aussi trouver la constante c de l'ellipse :

$$c = \frac{l_z^2}{G\mu^2 M} = 5708174 \ [\text{m}] \quad (18)$$

Puis, on cherche l'excentricité

$$\varepsilon = \frac{c}{r_{min}} - 1 = 0,1416 \quad (19)$$

L'énergie totale du système sera donc donnée par :

$$E = \frac{G^2\mu^3 M^2}{2lz^2}(\varepsilon^2 - 1) = -1,994 \times 10^{35} \ [\text{J}] \quad (20)$$

À l'aide de ε et de la constante c, on peut trouver le demi grand axe de l'ellipse (a) :

$$a = \frac{c}{1-\varepsilon^2} = 5824968 \ [m] \quad (21)$$

Finalement, à l'aide de cette dernière valeur, on peut trouver la période à l'aide de la troisième loi de Kepler :

$$T = 2\pi\sqrt{\frac{a^3}{GM}} = 140,69 \ [s] \quad (22)$$

Il est ensuite possible de mesurer ces différents paramètres à l'aide du programme en lui imposant les mêmes conditions initiales (équations 15). Pour mesurer la période, il suffit de mesurer le temps entre le début de la simulation et le moment où m_2 retrouve sa position initiale. Il est aussi très simple de mesurer le périapse et l'apoapse à l'aide du programme (distance minimale et maximale de la trajectoire obtenue). On peut ainsi mesurer la constante a ($a = \frac{r_{min}+r_{max}}{2}$).

Pour mesurer l'énergie du système, on calcule la moyenne de toutes les énergies à chacun des moments ($E = T + U$). De la même manière, pour mesurer le moment cinétique du système, on calcule la moyenne de tous les moments cinétiques ponctuels à chaque intervalle de temps (chaque seconde) qui s'obtiennent à l'aide de l'équation :

$$l_z = \mu(\vec{r} \times \vec{v}) \quad (23)$$

On utilise ensuite cette dernière valeur pour trouver la constante c à l'aide de l'équation 18. Finalement, pour mesurer l'excentricité, on calcule la moyenne entre les trois valeurs obtenues par l'équation 19 et les deux équations suivantes :

$$\varepsilon = 1 - \frac{c}{r_{max}} \quad (24)$$

$$\varepsilon = \sqrt{1 - \frac{c}{a}} \quad (25)$$

On obtient donc les paramètres suivants :

$$\begin{cases} l_z = 8,95 \times 10^{36} \ [\text{kg}\cdot\text{m}^2/\text{s}] \\ E = -2,065 \times 10^{35} \ [\text{J}] \\ c = 5638494 \ [\text{m}] \\ \varepsilon = 0,1425 \\ a = 5758346 \ [\text{m}] \\ T = 137 \ [\text{s}] \end{cases} \quad (26)$$

Donc, en comparant les deux ensembles de résultats, on remarque que les données du programme se rapprochent beaucoup des valeurs théoriques. Ces résultats viennent confirmer l'efficacité de notre algorithme et viennent donc appuyer la légitimité de son utilisation dans la suite de notre analyse.

3.2 Résultats obtenus par Monte Carlo

3.2.1 Effet de la masse

Le premier paramètre que nous avons étudié est la masse moyenne initiale des planètes. Afin d'évaluer l'effet de la masse sur le système, nous avons fait des centaines d'essais sur un système ayant toujours les mêmes paramètres, mais une masse moyenne différente. Concernant les paramètres qui ont été fixés pour cette simulation, nous avons utilisé des simulations comportant 150 planètes, une vitesse moyenne de 25000 m/s et un

moment angulaire moyen de $1.2 \cdot 10^{30}$ kg·m^2/s. Ces paramètres sont constants pour toutes les simulations que nous avons effectuées. Ensuite, afin de recueillir nos données, nous avons fait varier la masse moyenne de $0.2M_T$ jusqu'à $8M_T$ en effectuant des sauts de $0.05M_T$ où M_T désigne la masse de la Terre. Pour chaque masse donnée, nous avons effectué cinq simulations (pour un total de 780 simulations) et avons défini les résultats pour une masse donnée comme la moyenne arithmétique des cinq simulations. Dans un premier temps, les résultats sur le nombre de planètes en orbite selon la masse moyenne sont présentés dans le graphique suivant :

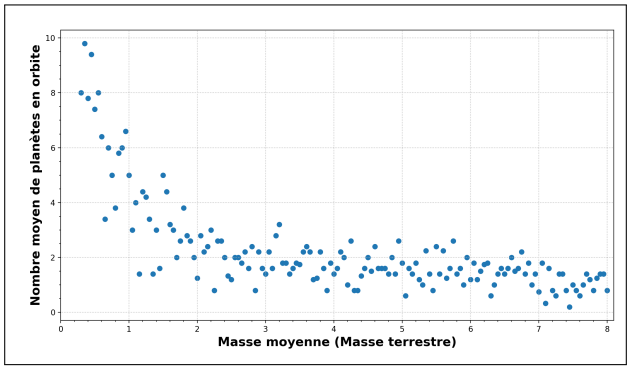

Figure 5 – Relation entre la masse moyenne des planètes et la stabilité des systèmes créés

On remarque d'abord que la stabilité des systèmes créés semble diminuer lorsque la masse moyenne de toutes les planètes augmente. On peut expliquer ce comportement en considérant le nombre de collisions se produisant en fonction de la masse : le rayon de chaque planète est fixé selon la masse et les planètes sont toujours placées initialement à l'intérieur d'une région dont l'aire est constante, donc, plus la masse augmente moins les planètes ont d'espace ce qui favorise les collisions. En effet, si l'on observe comment le nombre de planètes restant à la fin de la simulation varie en fonction de la masse moyenne, on s'aperçoit que plus la masse est grande, moins il reste de planètes à la fin de la simulation. On voit aussi clairement à la figure 6 que le nombre de planètes restant varie de la même façon que le nombre de planètes en orbite selon la masse.

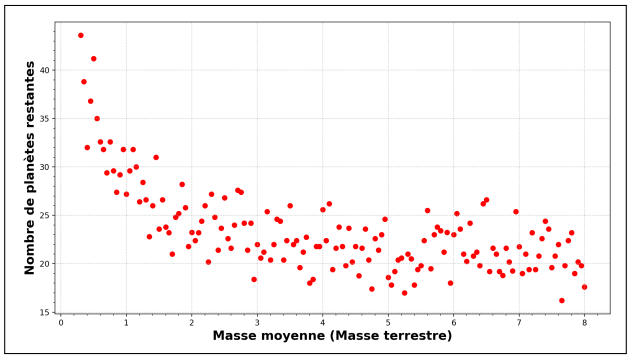

Figure 6 – Relation entre la masse moyenne des planètes et le nombre de planètes restant

On conclut donc que, plus la masse est grande, plus il y a de collisions, ce qui est défavorable à la formation d'un système stable, et donc que, plus la masse augmente (dans le cas de notre simulation), moins le système aura tendance à être stable.

3.2.2 Effet du nombre initial d'objets

Le second paramètre étudié pour évaluer son impact sur la stabilité des systèmes est le nombre initial de planètes. Comme pour la masse moyenne, quelques centaines de tests ont été effectuées. Les paramètres de la masse moyenne, de la vitesse moyenne et du moment angulaire moyen ont été respectivement fixés à $3M_T$, 2500 m/s et $1.2 \cdot 10^{30}$ kg·m^2/s. Nous avons fait varier le nombre de planètes entre 50 et 200 en effectuant une fois de plus 5 simulations pour chaque nombre de planètes. La relation entre le nombre initial de planètes et la stabilité des systèmes est présentée dans le graphique suivant :

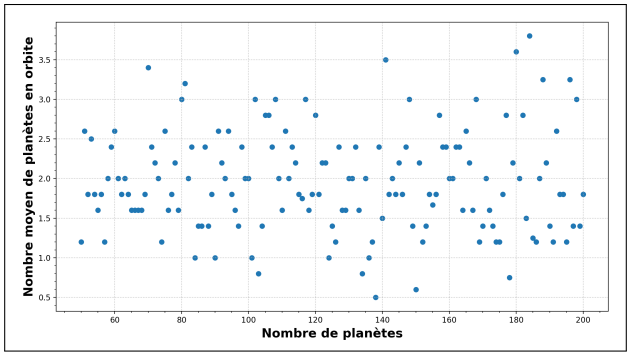

Figure 7 – Relation entre le nombre de planètes initial et la stabilité des systèmes créés

Ici, on remarque que le nombre de planètes au départ ne semble aucunement influencer la stabilité. En effet, aucune corrélation ne peut être tirée à partir de ces données. Ce résultat peut paraître surprenant et n'est pas nécessairement évident à prédire. Intuitivement, il serait logique de penser que la stabilité des systèmes augmente si on augmente le nombre de planètes initialement (la stabilité étant définie comme le nombre de planètes orbitant autour de la planète la plus massive). Il pourrait effectivement être possible de penser qu'il soit plus probable d'avoir des systèmes stables s'il y a plus de planètes au départ. Par contre, ce n'est pas ce qui est observé dans nos résultats, le nuage de point ne présente aucune corrélation claire.

De plus, nous pouvons observer le nombre de planètes restantes à la fin de la simulation en fonction du nombre initial de planètes. Cette relation est présentée à la figure 8.

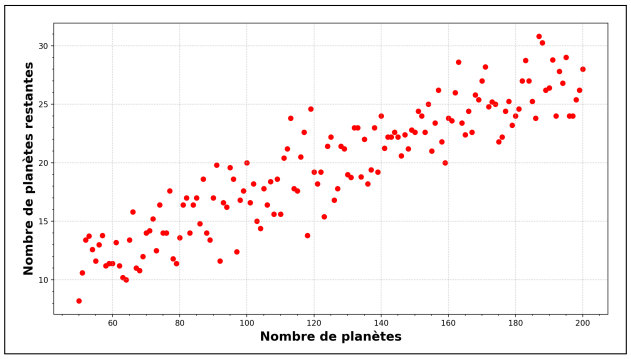

Figure 8 – Relation entre le nombre de planètes initiales et le nombre de planètes restantes à la fin de la simulation

On voit sur cette figure que le nombre de planètes à la fin de la simulation augmente à mesure que l'on augmente le nombre de planètes initial. Ce comportement peut paraître relativement prévisible et trivial. Or, ce qui est plus intéressant est que, même s'il existe une corrélation entre le nombre de planètes restant à la fin de la simulation et le nombre initial de planètes, il n'y a aucune corrélation entre le nombre de planètes et la stabilité du système. Nous concluons donc que, dans ce cas, il n'existe aucune corrélation entre le nombre de planètes restants à la fin de la simulation et la stabilité du système, ce qui est tout à fait à l'opposé du comportement observé pour la variation selon la masse moyenne où la corrélation était directe.

De plus, nous pouvons observer la variation de la masse de la planète centrale selon le nombre initial de planètes à la figure 9.

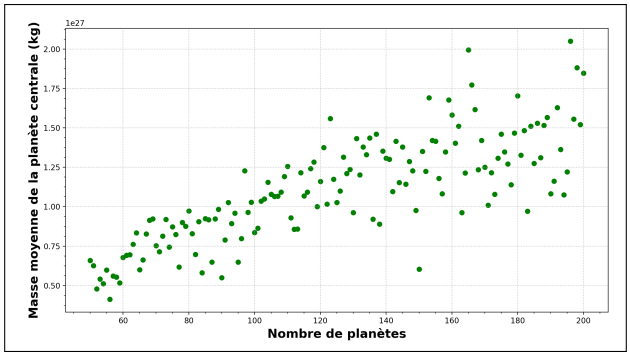

Figure 9 – Relation entre le nombre de planètes initial et la masse moyenne de la planète centrale créée

On voit sur cette figure que, de manière générale, la masse de la planète centrale augmente à mesure que le nombre de planètes augmente. Ce qui est relativement prévisible considérant que, plus il y a de planètes, plus il risque d'y avoir de collisions. Or, même si la planète centrale est plus lourde, et par conséquent, que son champ gravitationnel est plus intense, cela ne semble avoir aucune influence sur la stabilité du système.

En résumé, en faisant varier le nombre de planètes initial, on s'aperçoit que ce paramètre ne semble avoir aucune influence sur la stabilité du système à l'intérieur de l'intervalle étudié, et ce, même si le nombre initial de planètes influence le nombre de planètes final et la masse de la planète centrale.

3.2.3 Effet de la vitesse moyenne

Le dernier paramètre étudié est la vitesse moyenne initiale. Pour étudier son influence sur la stabilité des systèmes créés, nous avons fait varier ce paramètre en laissant les autres constants (comme pour les autres simulations). La masse moyenne a été fixée à $3M_T$, le nombre de planètes à 150 et le moment angulaire moyen a cette fois-ci été laissé aléatoire (l'orientation de la vitesse des planètes

est aléatoire). Nous avons fait varier les vitesses entre 0 m/s et 300 000 m/s avec des bonds de 5000 m/s. Dans ce cas, nous avons aussi effectué cinq simulations par vitesse donnée et défini les données associées à une vitesse comme la moyenne des cinq essais. Les résultats de la relation entre la stabilité et la vitesse moyenne initiale sont présentés dans le graphique suivant :

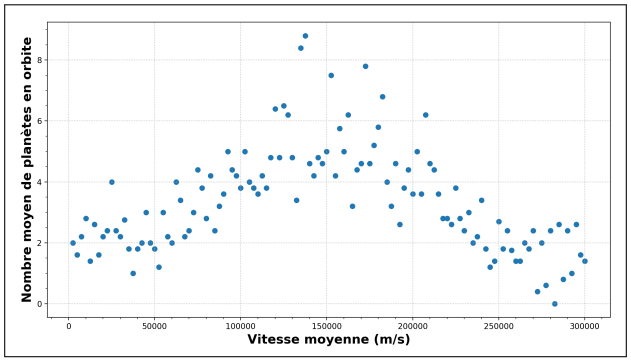

Figure 10 – Relation entre la vitesse moyenne initiale et le nombre de planètes en orbite stable au final

Ici, on remarque bien que si la vitesse moyenne initiale est trop petite ou trop grande, cela ne favorise pas la création de systèmes stables. Or, on remarque qu'au milieu du graphique (environ à 150 000 m/s), on obtient un maximum de stabilité. D'une part, pour expliquer l'instabilité à des petites vitesses, on peut penser que les masses n'ont, en général, pas assez d'énergie cinétique au départ pour soit s'évader du système ou créer une orbite elliptique. Ainsi, cela favorise les collisions du système. D'ailleurs, cette hypothèse est confirmée par le graphique suivant qui décrit la relation entre le nombre de planètes restantes à la fin de la simulation et la vitesse moyenne initiale :

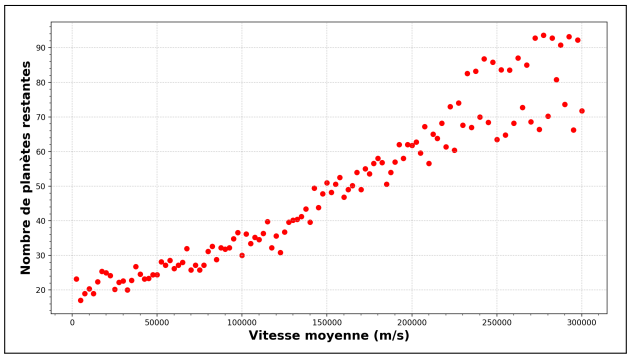

Figure 11 – Relation entre la vitesse moyenne initiale et le nombre de planètes à la fin de la simulation

On remarque donc que, si les planètes ont une faible vitesse au départ, il y aura plus de collisions et moins de planètes à la fin, ce qui cause l'instabilité du système.

D'autre part, si la vitesse moyenne des planètes du système au départ est trop grande, plusieurs d'entre elles ont certainement une vitesse plus grande que la vitesse de libération du système global. Ainsi, la probabilité de créer un système stable est plus faible. En ce sens, le graphique suivant présente la relation entre la masse de la planète centrale et la vitesse moyenne initiale.

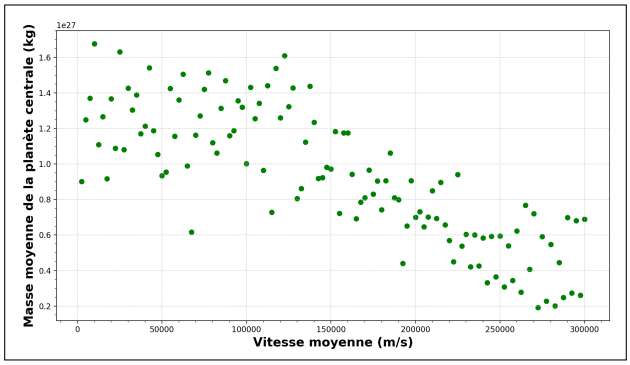

Figure 12 – Relation entre la vitesse moyenne initiale et la masse moyenne de la planète centrale créée

Bien que la corrélation soit faible, on remarque que, plus la vitesse moyenne des planètes augmente, moins la planète centrale est massive. Ainsi, une grande vitesse initiale ne favorise pas la création d'une planète centrale lourde (ce qui va dans le

même sens que la figure 11, puisque moins il y a de collisions, moins les planètes seront lourde en moyenne). Par le même fait, comme la planète centrale est moins massique, la vitesse de libération de cette planète sera plus faible, ce qui joue en défaveur de la stabilité, sachant que, initialement, la vitesse moyenne était élevée.

Bref, on peut conclure que la vitesse initiale moyenne des planètes favorise la création de systèmes stables dans la mesure où elle n'est ni trop grande ni trop faible, mais à mi-chemin entre les deux. Bien sûr, cette vitesse optimale varie selon les autres paramètres initiaux fixés. Dans notre cas, cette vitesse était autour de 150 000 m/s.

4 Conclusion

En conclusion, le but de notre projet était de créer un algorithme qui est en mesure de simuler l'interaction entre N corps. Une fois cet algorithme conçu, il a été utilisé pour évaluer les paramètres initiaux qui permettent la création d'un système stable. Avant de commencer l'étude des paramètres qui favorisent l'émergence de systèmes stables, nous devions vérifier l'efficacité de l'algorithme. C'est ce qui a été fait à partir d'un cas simple, soit un cas pour lequel il existe une solution analytique avec laquelle nous pouvions comparer nos résultats. En ce sens, l'algorithme a été utilisé pour étudier un problème à deux corps et a été comparé aux valeurs théoriques calculées. Ce test a validé l'efficace de l'algorithme et a donc permis son utilisation pour le but principal du projet : *une étude Monte Carlo de la formation de systèmes planétaires stables.*

Donc, nous avons fait varier trois paramètres initiaux jugés importants lors de plusieurs centaines de tests. Tout d'abord, le premier paramètre qui a été étudié est la masse moyenne des planètes. Après environ 780 simulations, les résultats obtenus ont permis de mieux comprendre l'effet de la masse. En effet, plus la masse moyenne des planètes est grande, plus le nombre de planètes en orbite autour de la masse centrale est petit. On obtient le même genre de conclusion pour le nombre de planètes libres à la fin de la simulation en fonction de l'augmentation de la masse moyenne. Ces deux résultats ont donc montré que plus la masse moyenne du système est grande, plus la probabilité d'avoir un système stable est petite.

Ensuite, le même genre de test a été utilisé pour comprendre l'effet du nombre initial de planètes dans la création de systèmes planétaires stables. En posant la masse, la vitesse et le moment angulaire constants, nous avons effectué plusieurs centaines de tests pour obtenir les conclusions qui suivent : plus le nombre initial de planètes est grand, plus le nombre de planètes libres et le masse la planète centrale augmentent. Or, malgré ces corrélations, le nombre initial de planètes ne semble avoir aucune influence sur le nombre de planètes en orbites. Ainsi, on remarque que le nombre de planètes initial ne semble pas avoir d'influence dans la création de systèmes stables.

Pour continuer, la dernière série de tests était pour évaluer l'effet de la vitesse dans les simulations. Avec la même méthodologie que précédemment, on obtient des résultats intéressants. Une petite vitesse nuit à la création de systèmes stables vu le manque d'énergie et une vitesse grande ne permet pas non plus d'obtenir des systèmes stables puisque plusieurs planètes sont tout simplement éjectées du système. Donc, pour remplir l'objectif de créer des systèmes stables, il faut une vitesse moyenne ni trop grande, ni trop petite (valeur qui devrait dépendre des autres paramètres initiaux).

En conclusion, à partir d'un algorithme de simulation à N corps, il a été possible de vérifier les paramètres initiaux qui influencent la création de systèmes planétaires stables à partir d'une méthode Monte Carlo. Les paramètres étudiés étaient la masse, le nombre de planètes initiales et la vitesse. Les résultats obtenus ont permis de mieux comprendre leur effet et l'intervalle de valeurs souhaitées en fonction des autres paramètres. D'ailleurs, ces résultats obtenus pourraient avoir une utilité dans un projet de plus grande envergure, comme une simulation de la création d'une galaxie complète. Quoiqu' intéressant, la complexité d'un tel problème dépasserait malheureusement le cadre de ce cours.

5 Références

[1] *La méthode de Verlet*, 2017. http://femto-physique.fr/omp/methode_de_verlet.php.

[2] Craig B. Agnor, Robin M. Canup & Harold F. Levison. On the Character and Consequences of Large Impacts in the Late Stage of Terrestrial Planet Formation. *Icarus*, 142 :219–237, 1999.

[3] Jonathan Dummer. *A simple time-corrected Verlet integration method.* http://www.lonesock.net/article/verlet.html.

Annexe

A Détails sur le programme utilisé

Le programme utilisé a entièrement été conçu dans le cadre de ce projet et n'utilise aucun programme préexistant sur le sujet. Afin de créer le programme, nous avons utilisé le langage de programmation Python, où les bibliothèques Numpy et Matplotlib ont été utilisées. Le programme au complet est composé de plusieurs centaines de lignes de codes réparties à l'intérieur de plusieurs fichiers. Il serait donc totalement inutile d'exposer en entier le programme utilisé à l'intérieur de ce document. Cependant, le programme peut être obtenu simplement en accédant à la page github du projet. Ainsi, le projet est disponible à l'addresse suivante : https ://github.com/FelixDesrochers/Gravitation

Les différents fichiers présents à cette page sont résumés ci-dessous :

1. Planet.py : fichier où les objets "planet" sont créés
2. code_efficace.py : fichier où l'intégration de Verlet est implémentée et les collisions gérées afin d'actualiser le système. L'animation est créée dans ce fichier aussi.
3. init_parametres.py : fichier où le système est créé à partir de paramètres initiaux donnés (nombre de planètes, vitesse moyenne, etc.)
4. Monte_Carlo.py : fichier permettant de rediriger les résultats de la simulation vers un fichier texte et où les paramètres initiaux sont définis
5. launch.sh : Fichier shell permettant d'automatiser le lancement du programme, soit, de répéter le programme en faisant varier les paramètres sans le relancer manuellement
6. Theorique.py : fichier indépendant pour effectuer les mesures nécessaires à la comparaison avec le modèle analytique