

UNIVERSITÉ DE MONTRÉAL

PHY 3075 – MODÉLISATION NUMÉRIQUE EN PHYSIQUE

**Projet 6 - Réseaux de neurones artificiels et détection du
boson de Higgs**

par :

Patrice Béchard

20019173

1 Introduction

En physique, les problèmes à N corps sont des situations où un très grand nombre d'agents interagissent entre eux tout au long de l'évolution d'un système. Ce genre de problème est courant en physique de la matière condensée, en traitant des électrons se déplaçant dans les réseaux d'atomes d'un solide, par exemple. Du côté de l'infiniment grand, la formation d'une galaxie peut aussi être interprétée comme un problème à N corps, où un très grand nombre d'étoiles interagissent entre elles pour former un regroupement ordonné.

Puisqu'il est inconcevable au niveau calculatoire de vouloir mesurer l'interaction de 10^{11} particules entre elles (interactions de l'ordre de $O(n^2)$), il est possible de simplifier grandement le problème en séparant le domaine sur lequel se produit la simulation en cases (particles in a box) et en mesurant l'interaction entre chaque case en fonction du nombre d'étoiles se retrouvant dans chacune de celles-ci. C'est précisément l'astuce qui sera utilisée pour la modélisation implémentée dans ce présent projet, où le but sera d'introduire un halo de matière noire à la galaxie simulée de sorte que la vitesse de rotation des étoiles suive le genre de vitesses observées expérimentalement sur les galaxies. Les détails quand à l'implémentation du problème sont présentés dans les notes de cours [1].

2 Validation du code

La première partie de la validation du programme consiste à s'assurer qu'une particule seule peut graviter autour d'un centre massif de façon plutôt stable et à peu près circulaire. Du coup, en généralisant à plusieurs particules, il est possible d'avoir la certitude que les particules individuelles auront elles aussi une trajectoire à peu près circulaire. Le résultat de cette première validation est présenté à la figure 1.

Comme attendu, cet orbite est très stable, signe que le programme fonctionne comme voulu. La prochaine étape de vérification consiste à implanter des vitesses initiales ne permettant pas à l'étoile de tracer un orbite képlérien circulaire. L'orbite tracé par une telle étoile ne devrait donc plus être circulaire, mais devrait rester plutôt stable dans le temps.. Le résultat de cette seconde validation est présenté à la figure 2

Encore une fois, le résultat obtenu est celui qui était attendu. Le programme concocté semble donc fonctionné comme voulu. Finalement, pour s'assurer que les conditions initiales étaient belles et bien celles attendu pour le système complet (comportant 1×10^6 étoiles), la figure 4.16 des notes de cours a été répliquée. Celle-ci est présentée à la figure 3.

La figure obtenue est presque identique à celle présentée dans les notes de cours. Les courbes obtenues suivent celles attendues presque identiquement. Cependant, la figure présentée dans les notes de cours

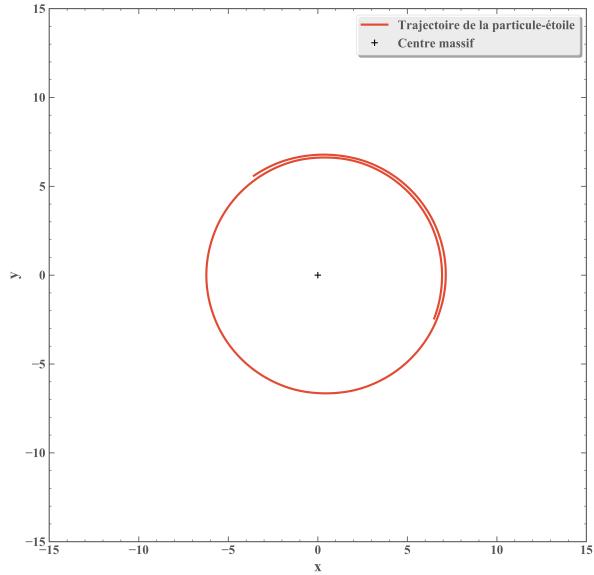


FIGURE 1 – Trajectoire d'une particule étoile gravitant autour d'un centre massif possédant une masse 1000 fois plus élevée que celle de la particule étoile. La vitesse initiale de la particule est choisie de sorte que l'orbite tracé par la particule soit circulaire. L'orbite tracé par la particule étoile est très stable et plutôt symétrique.

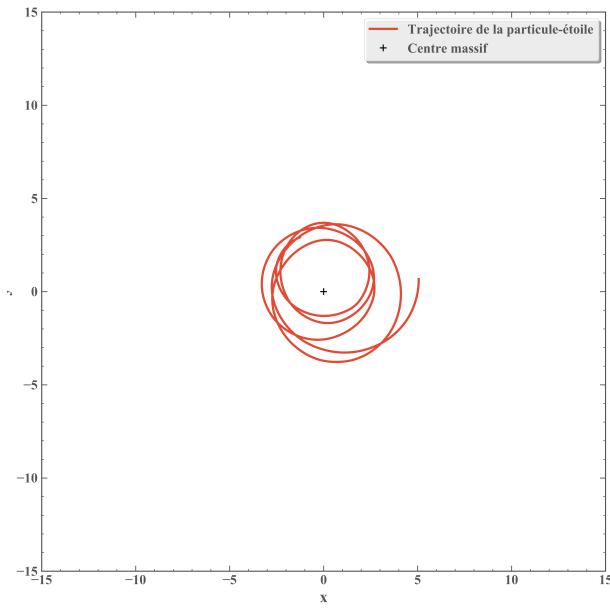


FIGURE 2 – Trajectoire d'une particule étoile gravitant autour d'un centre massif possédant une masse 1000 fois plus élevée que celle de la particule étoile. La vitesse initiale de la particule est choisie avec une composante aléatoire, de sorte que l'étoile ne possède pas un orbite circulaire. L'orbite tracé par la particule étoile est moins stable que l'orbite tracé à la figure précédente.

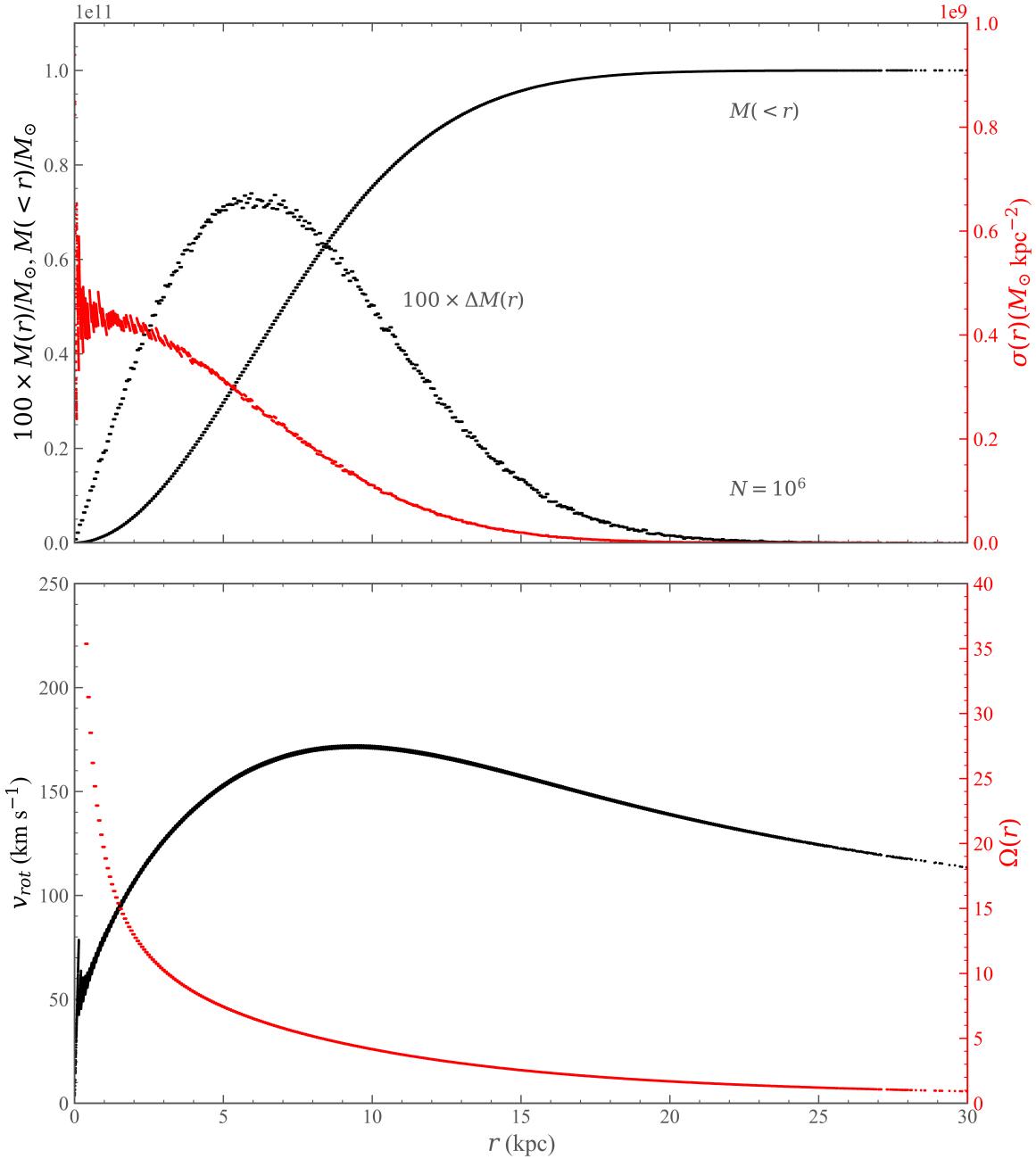


FIGURE 3 – Conditions initiales en quasi-équilibre képlérien pour une simulation de $N = 1 \times 10^6$ particules. La partie du haut montre la distribution radiale de la masse (noir) et de la densité surfacique (rouge), telle que produite par la procédure décrite dans les notes de cours. La partie du bas montre le profil radial de la vitesse de rotation (noir) et de la vitesse angulaire (rouge), correspondant à un profil képlérien en équilibre. Les paramètres définissant cette condition initiale sont données dans les notes de cours à la figure 4.15.

présente une moyenne par anneau des valeurs, alors que la figure présentée dans ce rapport montre plutôt les informations pour chaque étoile. C'est cette petite différence qui explique le bruit lorsque les valeurs se rapproche d'un rayon nul. Cela dit, les conclusions tirées sont les mêmes.

3 Résultats et discussion

Maintenant que le code a été vérifié, il est temps de se lancer dans le vif du sujet. Quatre simulations ont été effectuées, possédant des vitesses initiales képlériennes ou avec une composante aléatoire, et possédant ou non un profil de matière noire ajouté. Chacune des simulations possédait une configuration initiale des positions suivant un profil gaussien 2-D de rayon 4. Les simulations ont été perpétrées pour 8000 itérations, avec un pas de temps de 1×10^{-4} . Pour toutes les simulations effectuées, le même germe a été utilisé pour générer les nombres pseudo-aléatoires, de sorte qu'il soit possible de comparer directement les simulation l'une à l'autre.

La première simulation effectuée comporte 1 million de particules étoiles réparties selon un profil gaussien en deux dimensions. Les vitesses initiales de ces étoiles ont tout d'abord été assignées comme étant képlériennes, soit sur un orbite circulaire initialement. Cette simulation ne comporte pas encore de matière noire. Une évolution de la simulation sur trois pas de temps distincts est présentée à la figure 4, et une figure de la distribution des vitesses en fonction du rayon est présentée à la figure 5.

Sur la figure 4, le profil gaussien initial est toujours présent, mais une trou se forme rapidement en son centre. La simulation reste plutôt stable pour environ 1000 itérations. Pendant ce temps, la trajectoire des particules dont on suit le mouvement reste plutôt circulaire et stable. Par la suite, vers la moitié de la simulation (4000 itérations), le trou central est devenu plutôt grand et asymétrique. La trajectoire effectuée par les diverses particules n'est plus du tout circulaire, et des structures striées se forment en périphérie de la pseudo-galaxie. À la fin de la simulation (8000 itérations), alors que les particules ont presque complété un «orbite» (pas du tout circulaire), la galaxie présente quatre pseudo-bras possédant des trainées d'étoiles distribuées de façon plutôt asymétrique. La simulation n'a pas été poussée plus loin, puisque par la suite, les étoiles en périphérie de la galaxie ne sont plus assez attirées par le centre galactique et commencent à s'éloigner. Un nuage de particules non-gravitationnellement liées à la galaxie commence alors à se déplacer librement dans le domaine.

La figure 5 montre que la vitesse près du centre de la galaxie a grandement diminuée par-rapport à la vitesse donnée initialement. Cependant, la vitesse des particules en périphérie reste semblable à la vitesse initiale de ces particules.

La seconde simulation effectuée comporte encore une fois 1 million de particules étoiles réparties selon un profil gaussien en deux dimensions. Les vitesses des ces étoiles ont été assignées avec une composante

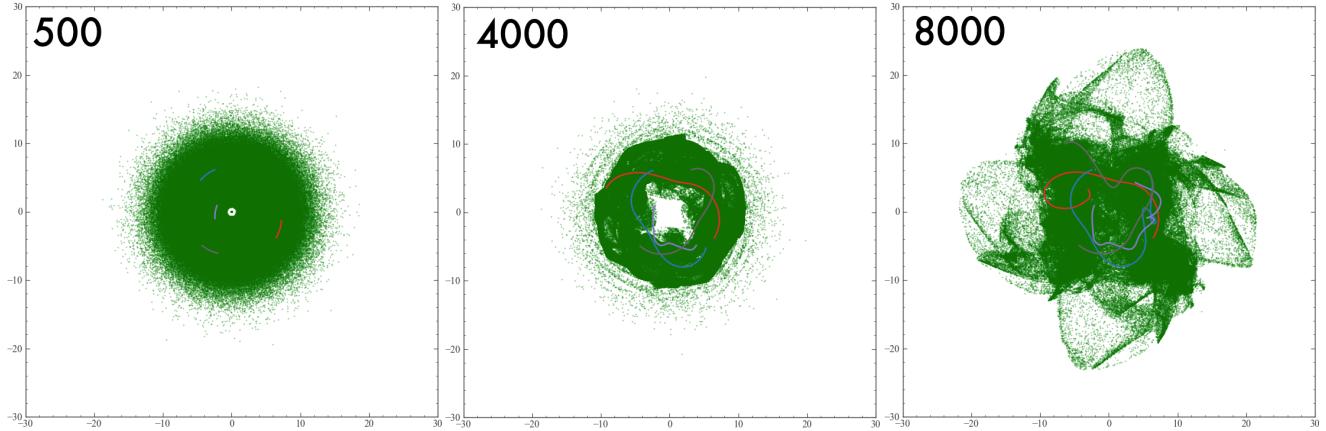


FIGURE 4 – Évolution d'une simulation comportant 1 million de particules étoiles avec vitesse initiale képlérienne et sans contribution de matière sombre.

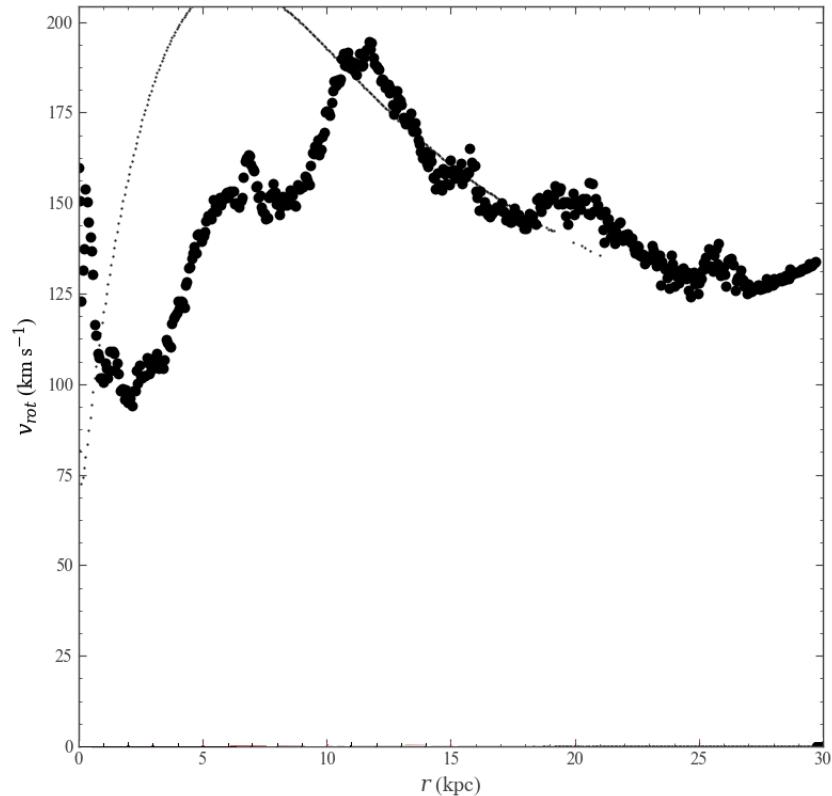


FIGURE 5 – Distribution des vitesses en fonction de la distance du centre du grillage pour la simulation avec vitesse initiale képlérienne et sans contribution de matière sombre. Les points fins représentent la distribution de vitesses initiale et les points gras représentent la distribution de vitesses à la fin de la simulation.

aléatoire, de sorte que leur orbite ne soit pas circulaire. Encore une fois, cette simulation ne comporte pas de matière noire. Des figures homologues à celles présentées pour la première simulation sont présentées aux figures 6 et 7.

Le début de la simulation est essentiellement identique à celui de la simulation précédente, comme le montre la configuration de la galaxie à première case de la figure 6. Au milieu de la simulation, un profil semblable est aussi obtenu. Cependant, les structures lignées obtenues avec les étoiles en périphérie n'est plus présente, signe que ces structures étaient issus de la vitesse identique reçue par les étoiles lors de l'initialisation. La trajectoire des particules suivies ne semblent cependant pas avoir changé pour les deux simulations, signe que celles-ci sont très similaires en résultat (on rappelle que le même germe est utilisé pour toutes les simulations pour pouvoir les comparer plus facilement). À la fin de la simulation, des bras spiraux beaucoup plus définis qu'à la simulation précédente semblent s'être formés. La trajectoire décrite par les particules suivies se sont aussi plutôt écartées de celles obtenues à la simulation où les vitesses étaient initialisées selon un orbite képlérien.

La distribution des vitesse obtenues en fonction de la distance du centre est plutôt étrange, montrant des pics de vitesses plus élevées à des rayons de 4kpc et 10kpc. Les vitesses en périphérie de la galaxie sont décroissantes plus les étoiles s'éloignent du centre, comme attendu lors de l'absence de matière sombre dans le système.

La troisième simulation effectuée est une réplique de la première, mais cette fois-ci, de la matière noire a été ajoutée au modèle. Le profil de matière noire utilisée dans la simulation est le profil de halo pseudo-isotherme, donné par l'équation suivante :

$$\rho(r) = \frac{\rho_0}{1 + \left(\frac{r}{r_c}\right)^2} \quad (1)$$

Pour l'ensemble des simulations pour lesquelles un halo de matière sombre a été ajouté, les paramètres utilisés étaient les suivants : $\rho_0 = 150M_*$, $r_c = 15$, $\alpha = 1.2$, où M_* représente la masse d'une particule étoile. Les figures 8 et 9 présentent l'évolution de la simulation à plusieurs pas de temps distincts, et la distribution de vitesses.

Encore une fois, dans cette simulation, les premières itérations ne semblent pas donner un résultat différent que ceux obtenus pour les simulations précédentes, alors que le profil général ainsi que la trajectoire des particules suivies sont les mêmes. La figure montrant la progression de la simulation à la moitié (4000 itérations) ne semble pas non plus différente de celle obtenue pour la première simulation effectuée, soit sans matière sombre et avec vitesses initiales képlériennes. Les structures lignées en périphérie de la galaxie sont toujours présentes, et le trou central possède la même forme. La figure à la fin de la simulation présente cependant une différence, soit celle que la galaxie semble s'être moins étalée dans le domaine,

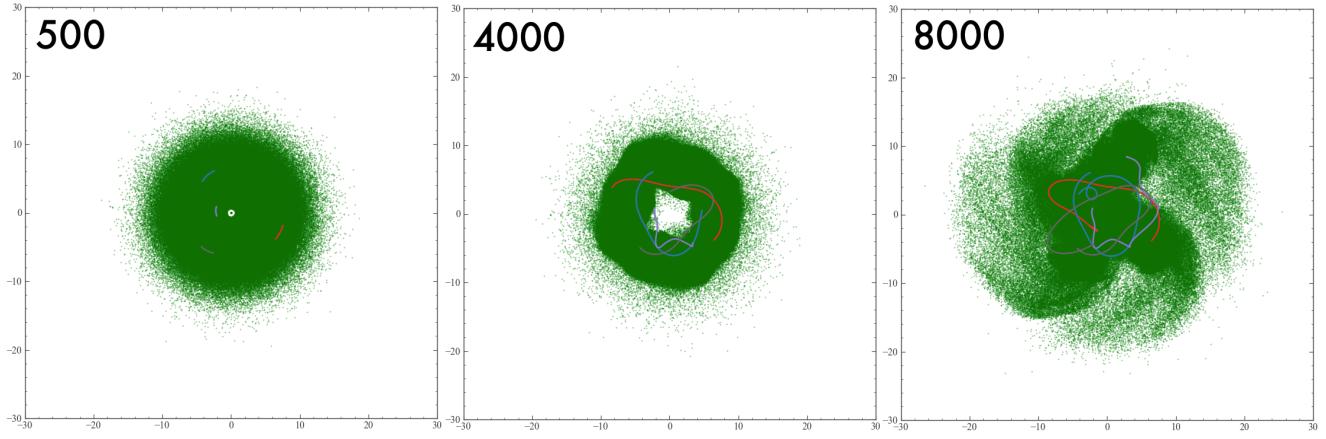


FIGURE 6 – Évolution d'une simulation comportant 1 million de particules étoiles avec vitesse initiale avec composante aléatoire et sans contribution de matière sombre.

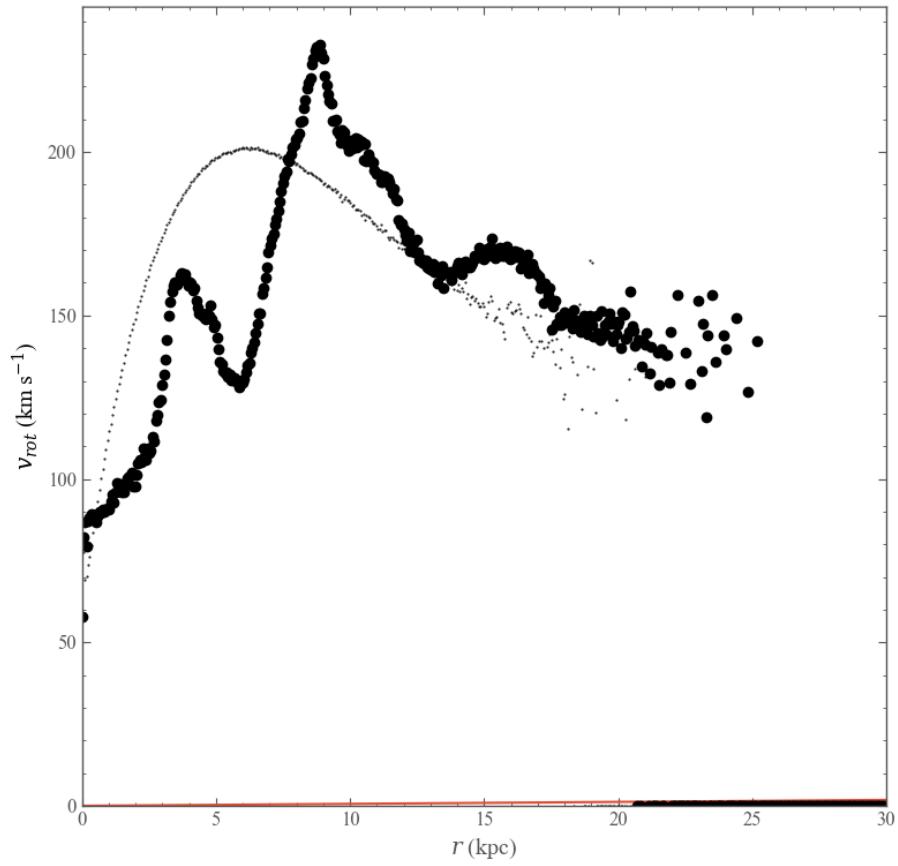


FIGURE 7 – Distribution des vitesses en fonction de la distance du centre du grillage pour la simulation avec vitesse initiale avec composante aléatoire et sans contribution de matière sombre. Les points fins représentent la distribution de vitesses initiale et les points gras représentent la distribution de vitesses à la fin de la simulation.

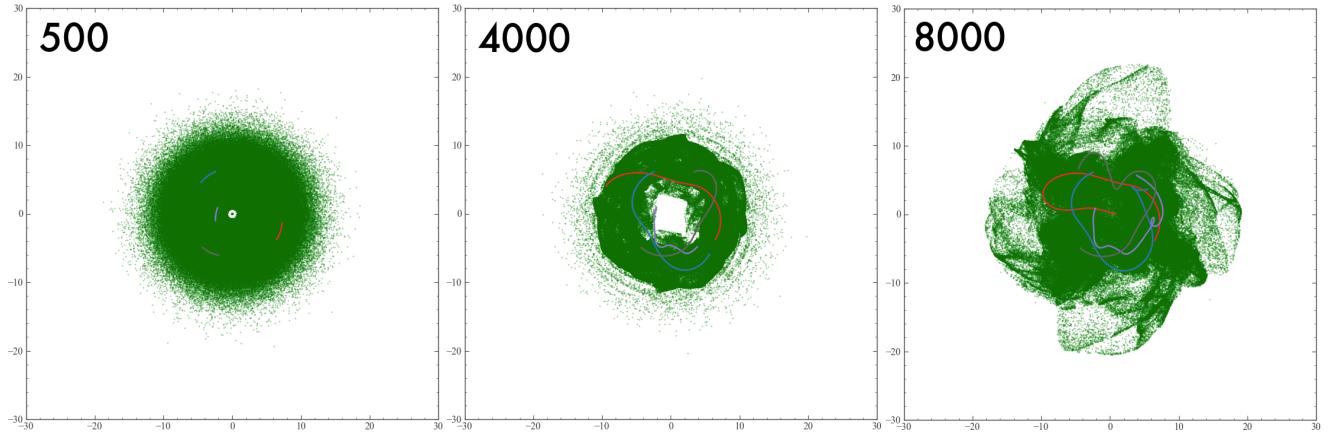


FIGURE 8 – Évolution d'une simulation comportant 1 million de particules étoiles avec vitesse initiale képlérienne et avec contribution de matière sombre.

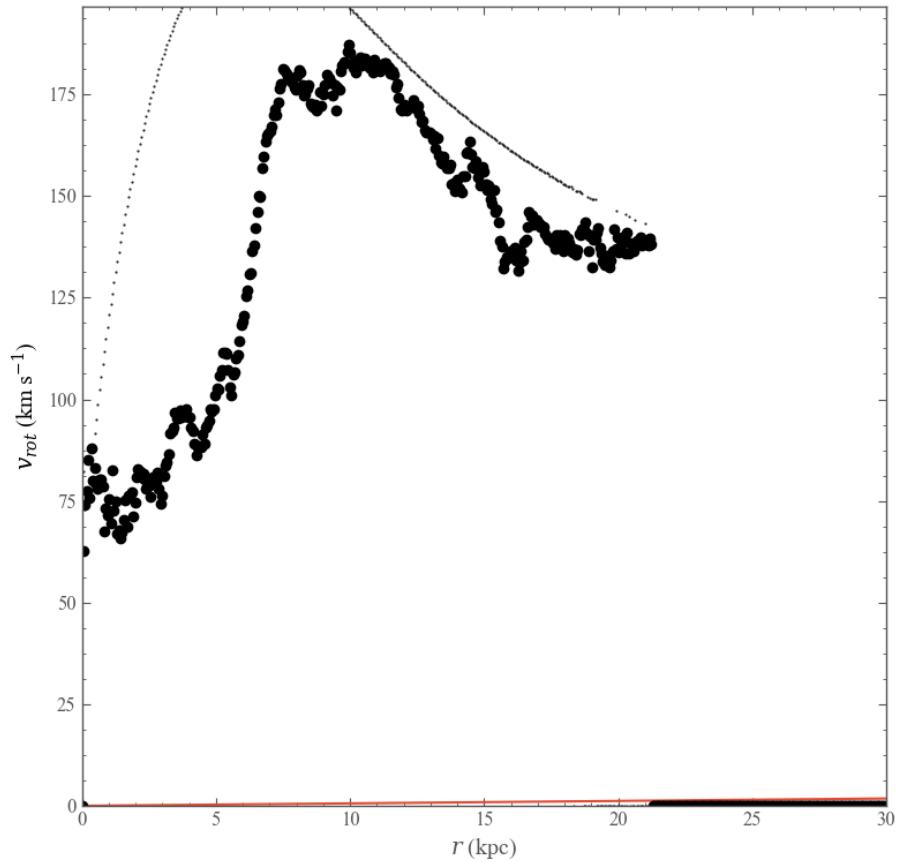


FIGURE 9 – Distribution des vitesses en fonction de la distance du centre du grillage pour la simulation avec vitesse initiale képlérienne et avec contribution de matière sombre. Les points fins représentent la distribution de vitesses initiale et les points gras représentent la distribution de vitesses à la fin de la simulation.

étant un peu plus compacte. Les mêmes petits bras sont cependant toujours visibles.

La distribution de vitesse obtenu pour cette simulation est semblable à celle obtenue pour la première simulation, alors que les particules près du centre possèdent des vitesses beaucoup plus petites que les vitesses initiales. Le confinement des particules plus près du centre de la galaxie est cependant visible sur cette figure, alors qu'il y a coupure à environ 22kpc puisqu'aucune particule étoile ne se trouve plus loin que cette distance à cette itération.

La quatrième simulation effectuée réplique la deuxième, mais avec l'addition de matière noire. Encore une fois, l'évolution de la simulation ainsi que la distribution de vitesses sont présentées aux figures 10 et 11.

La première case de la figure 10 est encore une fois identique à celles obtenues aux simulations précédentes. La deuxième case montre encore une fois un profil presque identique à la seconde case pour la deuxième simulation, soit celle possédant une composante aléatoire aux vitesses initiales données, mais ne contenant pas de halo de matière sombre. La troisième figure montre cependant que la galaxie a effectué environ 5 degrés de plus de révolution quant à la simulation 2, tout en montrant le même comportement général.

La différence principale entre ces deux simulations peut se voir sur la figure 11, alors que la vitesse des particules étoiles à un rayon plus loin que 17kpc est à peu près constante (en omettant le bruit dû à la faible quantité de particules étoiles plus la distance est grande). Cette caractéristique n'est évidente que sur cette simulation comportant la matière sombre effectuée, alors que les autres simulations présentent des vitesses qui diminuent plus les particules étoiles sont éloignées du centre galactique. Le reste de la distribution de vitesses est semblable à celui obtenu à la seconde simulation en ce qui a trait aux pics de vitesses plus élevées à 4kpc et 10kpc.

4 Exploration numérique

La première exploration numérique consiste à partir une simulation possédant cette fois-ci un profil elliptique au début. Le profil de matière sombre donné par l'équation 1 a été inclus dans la simulation. L'évolution de cette simulation est présenté à la figure 12.

Initialement, le profil elliptique commence à se distortionner et un trou se forme au centre de la galaxie, comme sur l'ensemble des autres simulations. Vers la moitié de la simulation (3500 itérations), deux grands bras spiraux symétriques sont présents, possédant chacun des trainées d'étoiles. La simulation continue alors d'évoluer, jusqu'à ce que système soit composé de deux entités séparées gravitant autour d'un point central et s'éloignant l'un de l'autre (5500 itérations). Les deux amas d'étoiles frappent ensuite les parois et rebondit, rendant la suite intéressante physiquement.

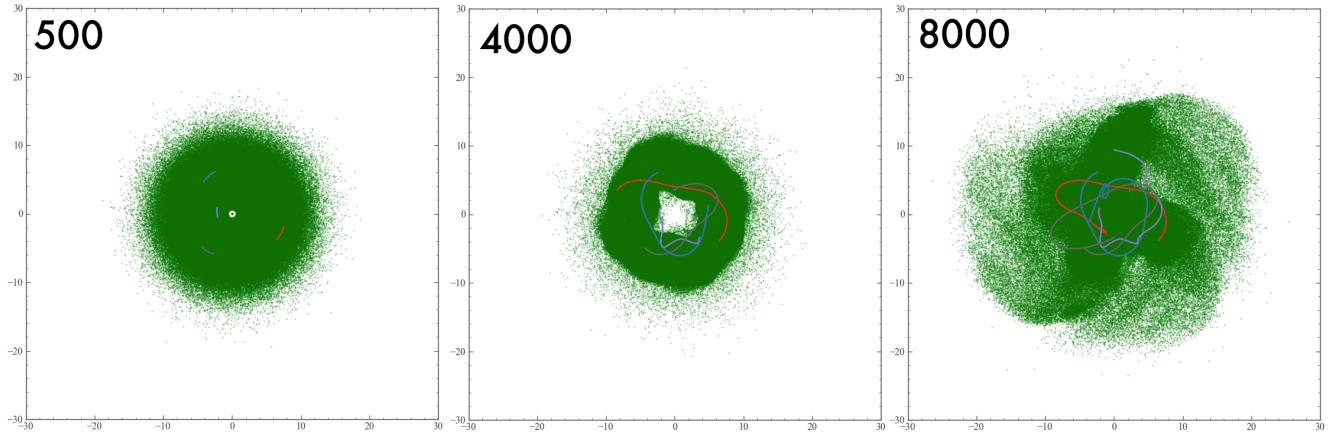


FIGURE 10 – Évolution d'une simulation comportant 1 million de particules étoiles avec vitesse initiale avec composante aléatoire et avec contribution de matière sombre.

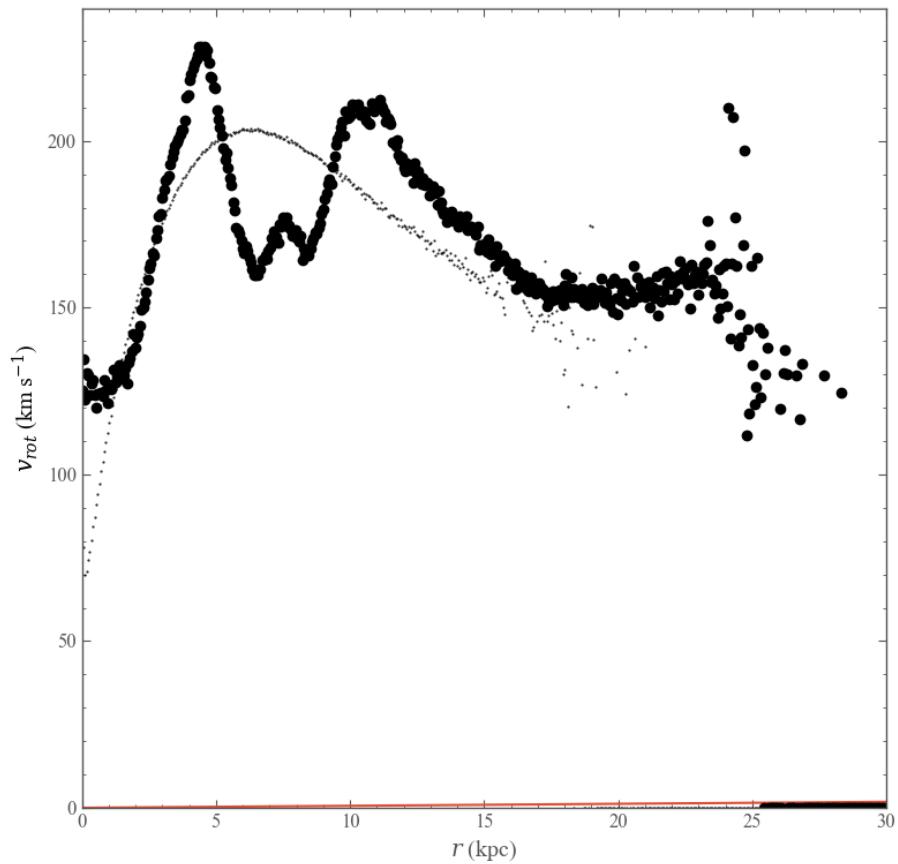


FIGURE 11 – Distribution des vitesses en fonction de la distance du centre du grillage pour la simulation avec vitesse initiale avec composante aléatoire et avec contribution de matière sombre. Les points fins représentent la distribution de vitesses initiale et les points gras représentent la distribution de vitesses à la fin de la simulation.

La seconde exploration a consisté à disposer les particules étoiles en croix pour observer l'évolution des bras spiraux. L'évolution de cette simulation est présentée à la figure 13.

Un trou au centre de l'amas d'étoile se développe tout d'abord lors des premières itérations, comme pour les autres simulations. Ensuite, ce trou devient très carré et symétrique (itérations 1500 à 3000) et des bras spiraux avec des traînées d'étoiles font leur apparition. Ceux-ci, deviennent de moins en moins important après la 3000e itération, jusqu'à ne plus être présents à la 5000e itération, où la distribution des étoiles ressemble plutôt à une rosace symétrique. Entre les itérations 5000 et 7500, le trou central s'effondre et la symétrie jusqu'ici présente dans la galaxie simulée disparaît et les étoiles forment alors un amalgame presque aléatoire de particules. Cependant, cette structure reste encore liée ensemble, et ce, à cause du halo de matière sombre introduit dans la simulation. Même en poussant le nombre d'itérations à 10000, la galaxie reste en un morceau. Il n'est cependant plus vraiment possible de pouvoir connaître la configuration initiale des étoiles rendu à ce point.

La distribution des vitesses à la fin de la simulation (présentée à la figure 14), n'est cependant pas comme celle obtenue pour la quatrième simulation effectuée (figure 11), alors que la vitesse des particules en périphérie de la galaxie restait constant en fonction de la distance. La distribution de vitesse obtenue suit essentiellement la distribution de vitesse initialement donnée au système, comme c'était le cas lorsqu'aucun halo de matière sombre n'était présent dans le système.

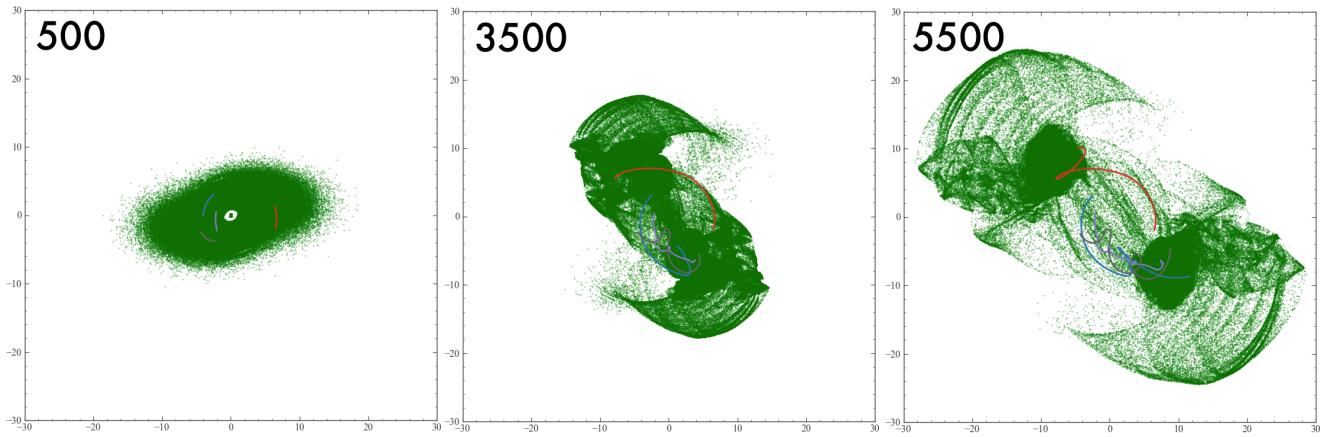


FIGURE 12 – Évolution d'une simulation comportant 1 million de particules étoiles initialement disposées selon un profil elliptique avec vitesse initiale avec composante aléatoire et avec contribution de matière sombre.

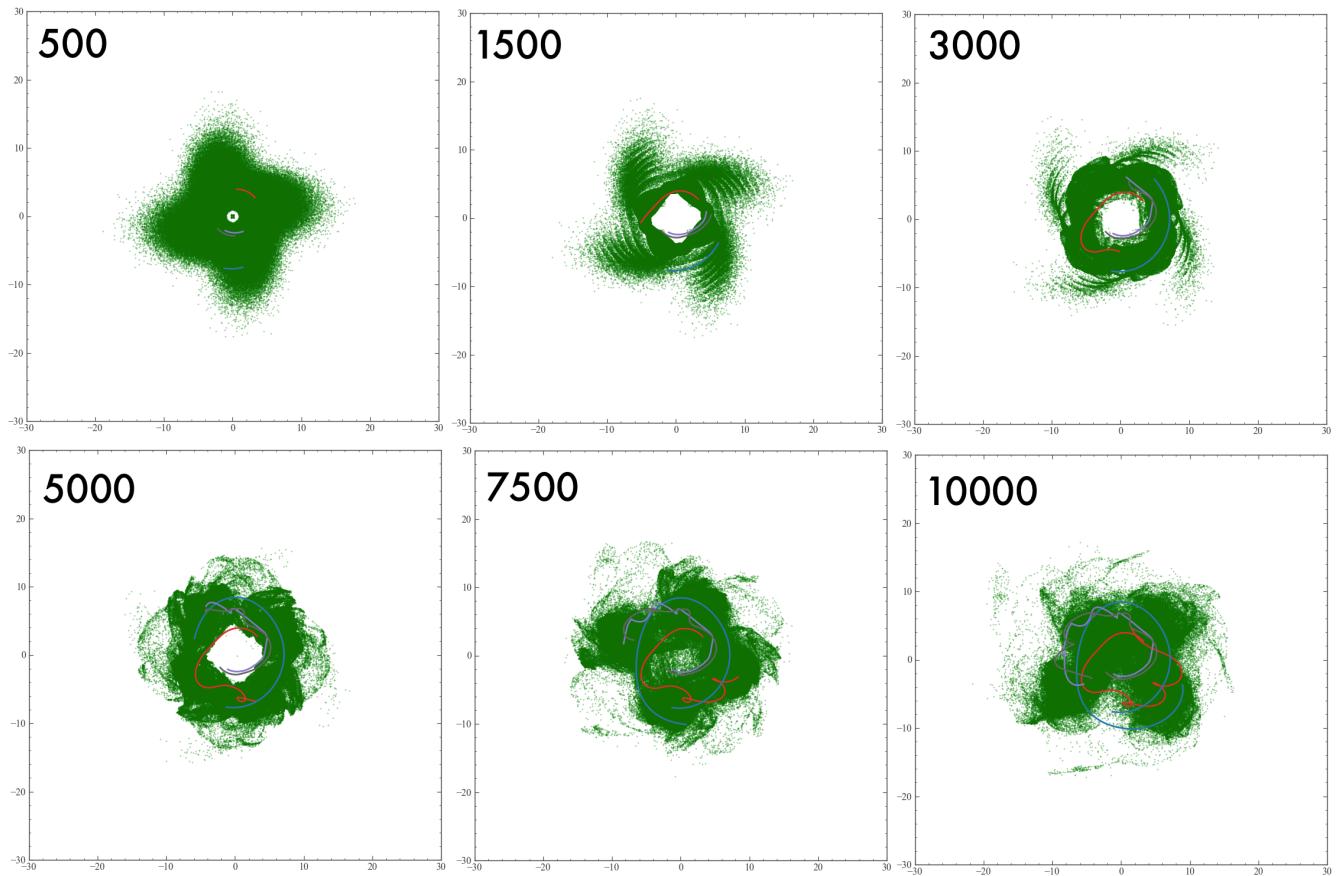


FIGURE 13 – Évolution d'une simulation comportant 1 million de particules étoiles initialement disposées selon un profil en croix avec vitesse initiale avec composante aléatoire et avec contribution de matière sombre.

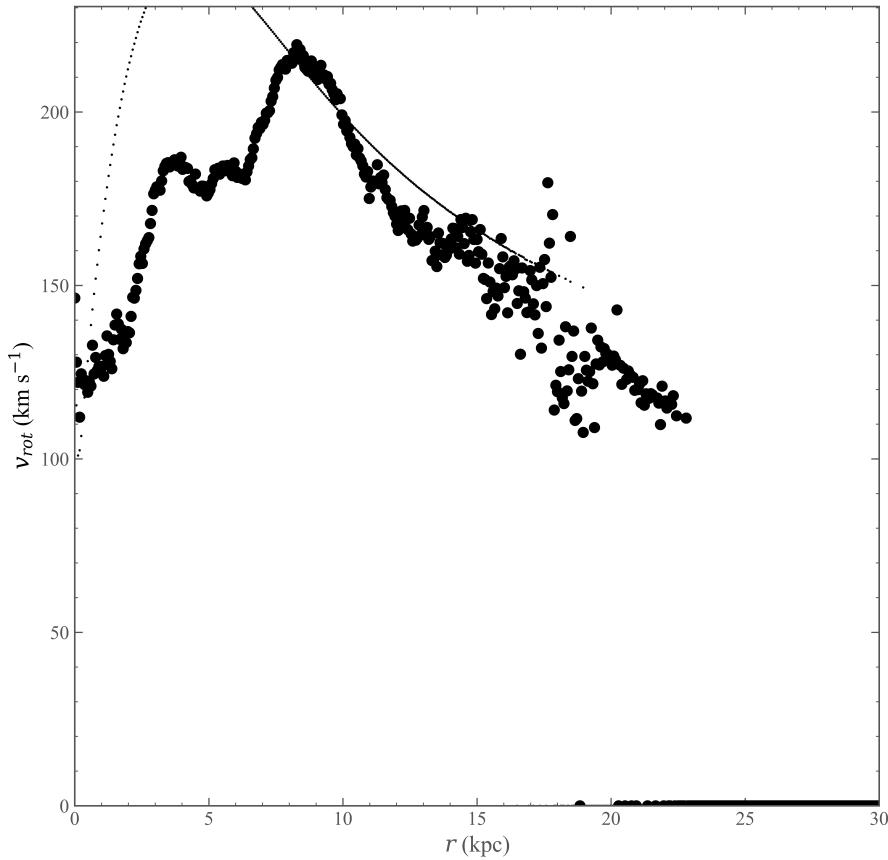


FIGURE 14 – Distribution des vitesses en fonction de la distance du centre du grillage pour la simulation possédant un profil initial en croix avec vitesse initiale avec composante aléatoire et avec contribution de matière sombre. Les points fins représentent la distribution de vitesses initiale et les points gras représentent la distribution de vitesses à la fin de la simulation.

5 Programme

```
# -*- coding: utf-8 -*-
# @Author: Patrice Bouchard 20019173
# @Date:   2017-04-01 23:38:41
# @Last Modified time: 2017-04-09 20:48:23
#
# Procedural version of Projet 4 PHY3075
#
#-----Modules-----
import numpy as np
import matplotlib.pyplot as plt
import sys
import time
from scipy.constants import pi as pi
import copy
import progressbar
from termcolor import colored

plt.style.use('patrice')
print("Execution Start Time : ", time.asctime())
start = time.time()           #time starts at beginning of execution

#-----Global Constants-----
GGRAV = 2.277e-7 #G in kpc**3/M_sun/P**2
N = 1000000 #number of star particles
RD = 4 #width of gaussian disk
MSTAR = 1.e11/N #masse mass of star particles
NA = 500 #number of rings
L = 30 #Domain from [[-L,L],[-L,L]]
#DR = 6*RD/NA #width of rings
DR = L/NA
M = 64 #number of cells
```

```

DELTA = 2*L / M #width of cells
dt = 0.0001 #time step
nIter = 10000 #number of iterations

rdmspeed = False #only change these lines
darkMatter = True
shape = 1 #0 : gaussian, 1: ellipse, 2: cross

if darkMatter:
    sigmaHalo = 150*MSTAR #dark matter parameters
else:
    sigmaHalo = 0
rHalo = 15
alpha = 1.2

```

#-----Fonctions-----

```

def init_pos():
    """Initial configuration of stars"""

    if shape == 0: #gaussian
        x = np.random.normal(0.,RD,N)
        y = np.random.normal(0.,RD,N)

    elif shape == 1: #ellipse
        x = np.random.normal(0.,RD,N)
        y = np.random.normal(0.,RD/2,N)

    elif shape == 2: #cross
        x1 = np.random.normal(0.,RD/2,N/2)
        y1 = np.random.normal(0.,RD,N/2)
        x2 = np.random.normal(0.,RD,N/2)
        y2 = np.random.normal(0.,RD/2,N/2)

        x = np.append(x1,x2)
        y = np.append(y1,y2)

    return x,y

```

```

def compute_mass_distr():
    sigmaDarkMatter = sigmaHalo/(1+(np.arange(NA)*DR/rHalo)**alpha)
    deltaM = np.bincount(inring,minlength=NA)*MSTAR #vectorized
    deltaM += 2 * pi * (np.arange(NA)*DR + 0.5*DR) * DR * sigmaDarkMatter
    Mr = np.cumsum(deltaM) #vectorized
    return deltaM,Mr

def init_speed(radius,inring,x,y):
    temp = np.arange(NA)
    temp[0] = temp[1]
    Omega = np.sqrt(GGRAV*Mr/np.power(temp*DR,3)) #index 0 set as nan
    Omega[0] = 0
    if rdmspeed: #keplerian or randomized speed
        vrot = Omega[inring] * radius
        vr = np.random.normal(0.,vrot*0.1,N)
        va = np.random.normal(0.,vrot*0.1,N) + 0.95*vrot
        vx = (-y*va + x*vr)/radius
        vy = ( x*va + y*vr)/radius
    else:
        vx = -Omega[inring] * y #- va * sin (angle)
        vy = Omega[inring] * x # va * cos (angle)
    return vx,vy

def compute_distance():
    """Computing the distance for the potential computation step"""
    bar = progressbar.ProgressBar(maxval=M+1, \
        widgets=[progressbar.Bar('=', '[', ']'), ' ', progressbar.Percentage()])
    bar.start() #progress bar for user

    dist = np.zeros([M+1,M+1,M,M]) #4-d array containing distances
    for i in range(M+1): #we compute only once instead of every iter
        bar.update(i+1)

```

```

for j in range(M+1):
    for k in range(M):
        for l in range(M):
            dist[i,j,k,l] = np.sqrt((i-k-0.5)**2 + (j-l-0.5)**2)

bar.finish()
return dist

def compute_dark_matter():
    """Compute the dark matter grid"""
    gridDarkMatter = np.zeros([M,M])
    for i in range(M):
        for j in range(M):
            gridDarkMatter[i,j] = np.sqrt((i-M/2)**2 + (j-M/2)**2)
    gridDarkMatter *= DR #we have the distance from the center for each cell
    gridDarkMatter = sigmaHalo / (1 + (gridDarkMatter/rHalo)**alpha)
    return gridDarkMatter

def compute_density(x,y):
    """Computing density of stars in each cell of the grid"""
    sigma = np.histogram2d(x,y,bins=[M,M],range=[[-L,L],[-L,L]]) #vectorized
    sigma = sigma[0] * MSTAR / (DELTA*DELTA) #only the histogram
    return sigma

def compute_potential(sigma,d):
    """
    Updating potential on each grid element
    with pre-calculated distance and density
    """
    pot = np.zeros([M+1,M+1])
    for i in range(M+1):
        for j in range(M+1):
            pot[i,j] = np.sum(sigma/d[i,j])

```

```

pot *= -GGRAV * MSTAR * DELTA
return pot

def compute_forces(x,y,pot):
    """Updating forces on each particle"""
    xs = (x % DELTA) / DELTA
    ys = (y % DELTA) / DELTA
    ix = np.array((x+L)/DELTA,dtype='int')
    iy = np.array((y+L)/DELTA,dtype='int')
    fx = -((pot[ix+1,iy]-pot[ix,iy])*(1-ys) + \
            (pot[ix+1,iy+1]-pot[ix,iy+1])*ys)/DELTA
    fy = -((pot[ix,iy+1]-pot[ix,iy])*(1-xs) + \
            (pot[ix+1,iy+1]-pot[ix+1,iy])*xs)/DELTA
    return fx,fy

def verlet(x,y,vx,vy,fx,fy,fx0,fy0):
    """Verlet algorithm for the evolution of the positions and speeds"""
    x += vx*dt + 0.5*fx0*dt*dt/MSTAR
    y += vy*dt + 0.5*fy0*dt*dt/MSTAR
    vx += 0.5 * dt * (fx0+fx)/MSTAR
    vy += 0.5 * dt * (fy0+fy)/MSTAR
    return fx,fy #assigned to fx0, fy0 at output

def adjust_position(x,y,vx,vy,radius,inring):
    """Adjust position of star if out of bounds"""
    xupper = (x > L) ; xlower = (x < -L) #bool arrays used to vectorize operations
    yupper = (y > L) ; ylower = (y < -L)
    x += xupper*(2*L - 2*x) + xlower*(-2*L - 2*x) #change position
    y += yupper*(2*L - 2*y) + ylower*(-2*L - 2*y)
    vx += xupper*(-2*vx) + xlower*(-2*vx) #change speed
    vy += yupper*(-2*vy) + ylower*(-2*vy)
    radius = np.sqrt(x*x + y*y) #update radius

```

```

inring = np.array(radius/DR,dtype='int') #update in what ring
return radius,inring

def show_fig417(x,y,i,xpos, ypos):
    """Recreation of figure 4.17"""
    plt.figure(1,figsize=(9,9))
    plt.plot(x,y,'go',ms=0.25)
    for j in range(len(xpos)):
        plt.plot(xpos[j],ypos[j],lw = 2)
    temp = -15
    """
    for i in range(self.M+1): #grid
        plt.axhline(temp,-15,15,c='r',lw=0.5)
        plt.axvline(temp,-15,15,c='r',lw=0.5)
        temp += self.DELTA
    """
    plt.axis([-30,30,-30,30])
    #plt.savefig("/Users/user/417.png",format='png', dpi=1000)
    plt.savefig("testfig/417_%05d.png"%i,format='png')
    plt.close()

def show_fig421(vrot0,vrot1,i=None):
    radiusplot = np.arange(NA)*30/NA
    vrot0 = vrot0*3.086e+16/7.5e15 #convert to km/s
    vrot1 = vrot1*3.086e+16/7.5e15

    plt.figure(2,figsize=(9,9))
    plt.plot(radiusplot,vrot0,'ko',ms=0.5)
    plt.plot(radiusplot,vrot1,'ko')
    plt.ylabel(r'$v_{\text{rot}}$ (km s$^{-1}$)', color='k')
    plt.xlabel(r'$r$ (kpc)')
    plt.axis([0,30,0,max(vrot1)*1.05])

```

```

if i != None:
    plt.savefig("testfig/421_%d.png"%(i),format='png')
else:
    plt.savefig("testfig/421.png",format='png', dpi=1000) #last one in high quality
    plt.close()

#-----Main-----
print("Initiating seed")
np.random.seed(1)

print("Initiating positions")
x,y = init_pos()

#initialize empty arrays
radius = np.zeros(N) ;
inring = np.zeros(N,dtype='int')
vx = np.zeros(N) ; vy = np.zeros(N)
fx0 = np.zeros(N) ; fy0 = np.zeros(N)
radius,inring = adjust_position(x,y,vx,vy,radius,inring) #set radius and in which ring

print("Computing mass distribution")
deltaM,Mr = compute_mass_distr()

print("Initiating speed")
vx,vy = init_speed(radius,inring,x,y)

distr = np.bincount(inring,minlength=NA) #used to plot speed distribution
vrot0 = np.zeros(NA)
vrot = np.sqrt(vx*vx + vy*vy)
for i in range(N):
    vrot0[inring[i]] += vrot[i]
for i in range(NA):

```

```

if distr[i] != 0:
    vrot0[i] /= distr[i]

print("Computing dark matter distribution")
gridDarkMatter = compute_dark_matter()

print(colored("Computing distance",'red'))
dist = compute_distance()

part2x = [[] for i in range(4)] #to track particles during simulation
part2y = [[] for i in range(4)]

for i in range(1,nIter+1):
    itertime = time.time()
    print("Iteration #",i)
    sigma = compute_density(x,y) #compute density
    sigma += gridDarkMatter #add dark matter contribution if any
    pot = compute_potential(sigma,dist) #compute potential
    fx,fy = compute_forces(x,y,pot) #compute forces
    fx0,fy0 = verlet(x,y,vx,vy,fx,fy,fx0,fy0) #verlet algorithm
    radius,inring = adjust_position(x,y,vx,vy,radius,inring) #update radius, in which ring
    print(time.time()-iterrime)
    if i%100 == 0:
        show_fig417(x,y,i,part2x,part2y) #position plot
        vrot = np.sqrt(vx*vx+vy*vy)
        vrot1 = np.zeros(NA)
        distr = np.bincount(inring,minlength=NA)
        for j in range(N):
            vrot1[inring[j]] += vrot[j]
        for j in range(NA):
            if distr[j] != 0:
                vrot1[j] /= distr[j]
        show_fig421(vrot0,vrot1,i) #speed distribution plot

```

```

for j in range(4):
    part2x[j].append(x[j]) #append current particles position
    part2y[j].append(y[j])

for i in range(len(inring)):
    if inring[i]>=500:
        inring[i] = 499

distr = np.bincount(inring,minlength=NA)
vrot = np.sqrt(vx*vx + vy*vy)
vrot1 = np.zeros(NA)
sigma1 = np.zeros(NA)

for i in range(N):
    vrot1[inring[i]] += vrot[i]

for i in range(NA):
    if distr[i] != 0:
        vrot1[i] /= distr[i]

show_fig421(vrot0,vrot1) #speed distribution

totaltime=time.time()-start
print("Total time : %d h %d m %d s"%(totaltime//3600,(totaltime//60)%60, totaltime%60))

```

Références

- [1] Charbonneau, P., *PHY3075 - Modélisation numérique en physique, Chapitre 4 : Problème à N corps*, Université de Montréal, Montréal, 2017, 36p.