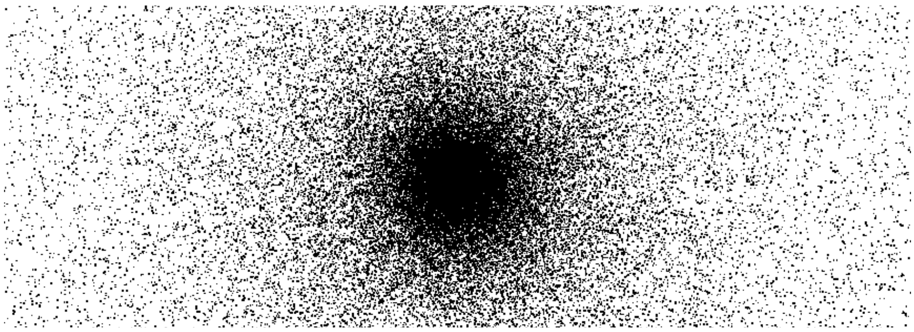
Galaxy Generation

Jugend Forscht 2018

Emile Hansmaennel `emile.hansmaennel@gmail.com`

9. Februar 2018



Das Ziel meines Projektes ist es, Realitätsgetreue Galaxien und Dunkle Materie Halos zu generieren. Hierzu verwende ich das sogenannte "Navarro-Frenk-White" Profil welches in Kombination mit der "Random Sampling" Methode die Dichteverteilung der Sternenpositionen in Koordinaten für einzelne Sterne umgewandelt.

Vergleicht man die generierten Galaxien mit echten Galaxien fällt auf das die Sterne sich anders verhalten. Dies lässt sich durch Dunkle Materie erklären, welche man jedoch nicht direkt beobachten kann. Es kann also nur aufgrund ihrer Auswirkungen auf andere Objekte auf sie geschlossen werden, weshalb es nicht ganz Trivial ist sie sichtbar darzustellen.

Im Verlauf des Projektes haben sich mir jedoch auch andere Teilbereiche eröffnet wie z. B. die Generation von Spiralgalaxien, die Optimierung von Rechenprozessen und die Nutzung von einem neuronalen Netz zur Anpassung der generierten Galaxie an eine reale Galaxie.

Inhaltsverzeichnis

1 Einleitung

Nach meinem letzten Jugend-Forscht Projekt ergab sich mir die Möglichkeit ein Praktikum im Zentrum für Astronomie in Heidelberg zu absolvieren. Über die Social-Media Plattform Reddit stellte ich den Kontakt mit Tim Tugendhat her der zurzeit seinen PhD. in Physik an der Universität in Heidelberg macht. Dieser ermöglichte es mir, die physikalische Fakultät an einer Uni mal genauer zu sehen und das täglich leben eines Physikers mitzuerleben.

Während des Praktikums stellte ich fest das ich die im letzten Jahr erlerne Fähigkeit mit Python[1] zu Programmieren und mit Blender[2] umzugehen nutzen konnte um Galaxien darzustellen. Dies war insgesamt unglaublich interessant und zeigte mir zum wiederholten mal: Projekte sind sehr dazu geeignet um sich in neues einzuarbeiten oder neues zu lernen und bieten einem ein Ziel welches man erreichen möchte was einem immer genügend Motivation bietet weiterzumachen.

Eine frage die ich mir öfters gestellt habe war, warum man eigentlich Galaxien simuliert? Wäre es nicht einfacher einfach in den Himmel zu gucken und die bereits bestehenden Galaxien zu beobachten? Nach kurzer Recherche lag die Antwort auf der Hand: Galaxien brauchen mehrere Millionen Jahre um sich zu entwickeln, also kann man ihre Entwicklung als normaler Mensch nicht in dem Umfang beobachten, um dann daraus Schlüsse zu ziehen. Daher simuliert man die Galaxien und kann dann somit vorhersagen oder herausfinden wie die Galaxien entstanden sind bzw. was mit ihnen passieren wird.

1.1 Themen

- Generierung von elliptischen Galaxien
- Generierung von einem Dark-Matter Halo um die elliptische Galaxie
- Stauchung und Streckung des Dark-Matter mit Beeinflussung der eigentlichen Galaxie
- Beschleunigung des Generierungsprozesses mithilfe einer sogenannten "lookup-table"
- Aufbau eines neuronalen Netzes für die unbeaufsichtigte Generation von Galaxien
- Generation von Spiralgalaxien

1.2 Motivation

Die Motivation für das Projekt kam praktisch direkt nach meinem letzten Jugend Forscht Projekt bei dem ich mich mit der Vorhersage von Satellitenkollisionen beschäftigt habe. Durch mein Praktikum im Zentrum für Astronomie der Uni Heidelberg kam ich auf die Idee, ich könnte mein Wissen im Bezug auf die Programiersprache Python und der 3D-Suite Blender mithilfe eines Projektes erweitern. Ein Projekt zu haben um sich an etwas Neues heranzuwagen ist sehr empfehlenswert wie ich schon in letzten Jahr gemerkt habe, ich hatte also wieder ein Projekt welches mich Tag für Tag motiviert hat etwas zu erreichen.

[1] Programmiersprache
[2] 3D Software Suite

2 Hauptteil

2.1 Generierung von elliptischen Galaxien

2.1.1 Das Navarro-Frenk-White Profil

Das Navarro-Frenk-White Profil (NFW-profil) ist ein Profil zur Simulation von Masseverteilungen in N-Körper-Simulationen. Im Grunde genommen bekommt man durch das Profil die Wahrscheinlichkeit das ein Stern in einem Abstand r vom Mittelpunkt der Galaxie existiert. Die Funktion die dies bewerkstelligt ist im Allgemeinen wie folgt aufgebaut:

$$\rho_{NFW}(r) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot \exp\left(\frac{-\phi(r)}{\sigma^2}\right) \tag{1}$$

$$\phi(r) = \frac{4\pi \cdot G \cdot f_0 \cdot R_s^3}{r} \cdot ln\left(1 + \frac{r}{R_s}\right)$$

Beispiel Werte:

$$\begin{aligned} sigma &= 200 \\ f_0 &= 0.1 \\ R_s &= 10000 \\ pi &= 3.141592 \\ e &= 2.718281 \\ G &= 4.302\text{e-}3 \end{aligned}$$

Möchte man herausfinden wie wahrscheinlich es ist das ein Stern generiert wird, setzt man den Abstand des Sternes vom Mittelpunkt der Galaxie in die Formel (1) ein. Möchte man z. B. wissen wie wahrscheinlich es ist, das ein Stern der die Koordinaten (x_1, x_2, x_3) besitzt generiert wird, wird der Abstand zum Mittelpunkt der Galaxie mithilfe des Satzes des Pythagoras berechnet:

$$r = \sqrt{{x_1}^2 + {x_2}^2 + {x_2}^2} \tag{2}$$

In dem Beispiel wird also der Wert r in das NFW-Profil gegeben:

$$\rho_{NFW}(r) = \cdots = s$$

Als Lösung erhält man einen Wert der in einem Intervall [$\rho(r_{min})$; $\rho(r_{max})$] liegt. Rechnet man $\rho(r_{min})$ und $\rho(r_{max})$ aus, ist es möglich den jeweiligen Ergebnissen anhand dieser Werte eine Wahrscheinlichkeit zwischen 0 und 100% zuzuordnen Sodas es möglich ist zu entscheiden, ob ein Stern bei $P(x_1|x_2|x_3)$ generiert werden soll oder nicht.

2.1.2 Random Sampling

Um jetzt herauszufinden ob der Stern bei $P(x_1|x_2|x_3)$ generiert wird, wird ein zufälliger Wert n im Intervall [$\rho(0)$; $\rho(r_{max})$] generiert und mit dem Wert x verglichen. Ein Stern wird generiert wenn gilt $n < x$. Wenn jedoch $n > x$ gilt wird kein Stern generiert. Dieser Prozess wird Random Sampling gennant und ist einer der Knackpunkte wenn es darum geht die Zeit in der ein Stern generiert wird zu reduzieren. Eine Möglichkeit dies zu tun sind sogenannten Lookuptabellen (siehe Sektion 2.1.3)

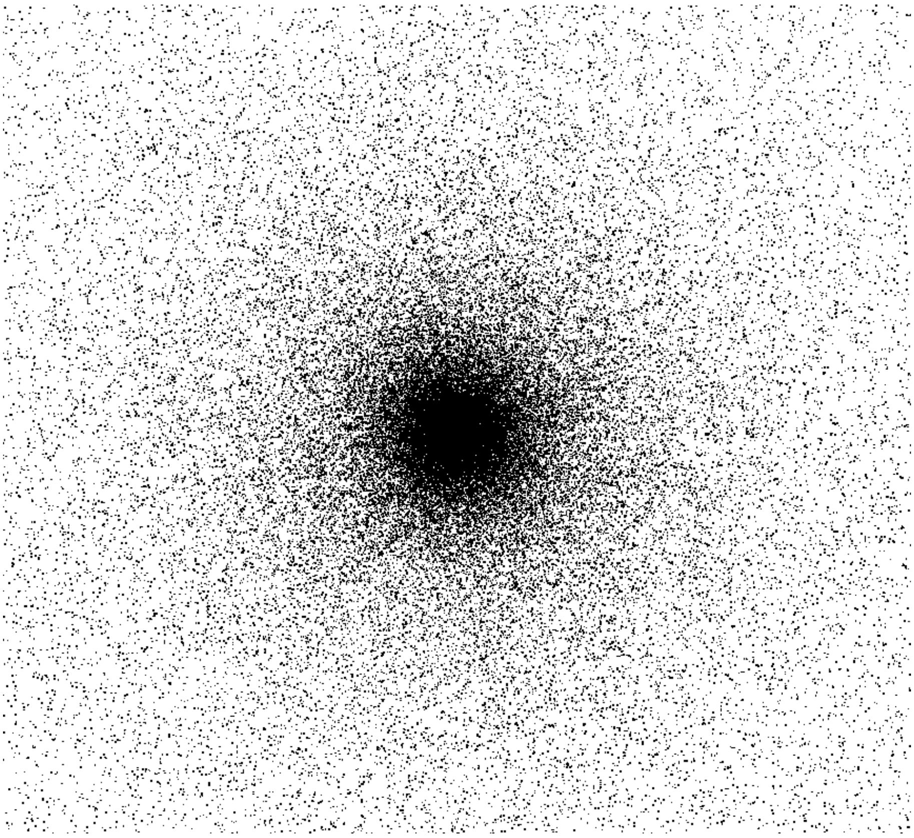


Abbildung 2: Eine mit dem NFW-profil und der Random Sampling Methode generierte Galaxie

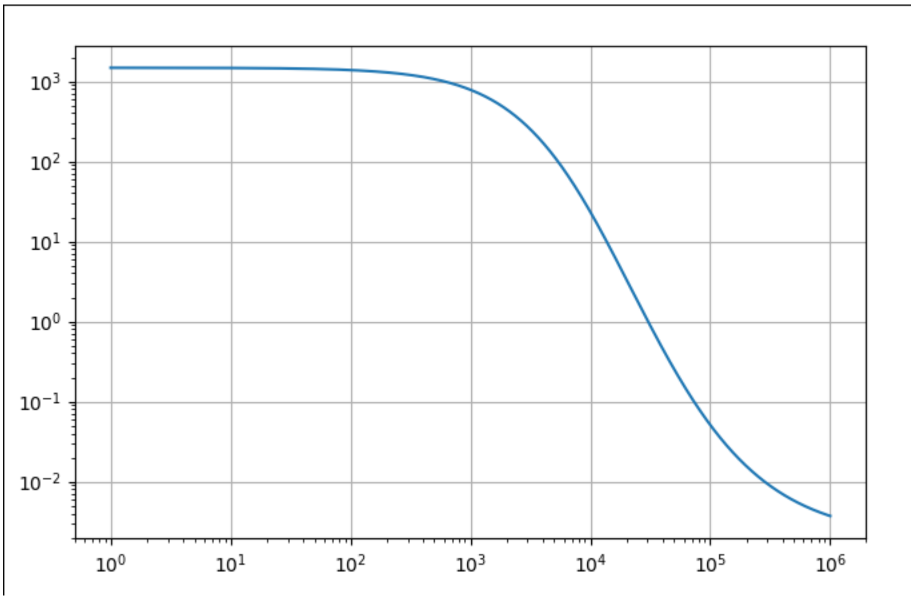


Abbildung 3: Die Rho Funktion im Intervall [0 ; 10^7] geplottet mithilfe von Logarithmischen Achsen.
Die x-Achse beschreibt die Entfernung zum Mittelpunkt der Galaxie
Die y-Achse beschreibt die Warscheinlichkeit das ein Stern generiert wird

2.1.3 Lookup Tabellen

Um das generieren zu Beschleunigen wird eine sogenannte “lookuptable” verwendet. (→ 2.5.1) dabei wird die Funktion aus dem NFW-profile (1) in eine Tabelle geschrieben die im .csv-format in eine Datei gespeichert. Dies hat den Vorteil das die Ergebnisse gespeichert vorliegen und somit für andere Berechnungen weiterverwendet werden können und bei der Berechnung in den Arbeitsspeicher geschrieben werden, wodurch die Ergebnisse aus der Funkion bei Bedarf vorliegen und nicht erst berechnet werden müssen.

2.2 Generierung eines Dunkle-Materie Halos durch Anpassung des NFW-Profils

Die Rotationskurve von Spiralgalaxien verhält sich in der Realität anders als in einer Simulation. Der Unterschied lässt sich durch eine Kraft erklären die Auswirkungen auf Materie haben kann, jedoch nicht sichtbar ist. Dadurch kann diese Kraft nur durch Rückschlüsse beschrieben werden. Verhält sich ein Objekt also nicht so, wie es aufgrund der sichtbaren auf es einwirkenden Kräfte tun sollte, so muss eine andere Kraft vorhanden sein, die das Objekt beeinflussen. Diese “Kraft” entsteht voraussichtlich aufgrund von dunkler Materie. Dadurch das man Dunkle Materie durch beobachten von Sternen orten kann indem man berechnet wie sich die Sterne theoretische verhalten sollten und dies mit den realen Gegebenheiten vergleicht kann man die Funktion die eigentlich die Dichte der Sternenverteilung (NFW-profil) erklären soll so anpassen das man die Dichte der Verteilung von dunkler Materie berechnen kann. Das NFW-profil (1) kann also so angepasst werden, dass es statt die Wahrscheinlichkeit das ein Stern generiert wird die Wahrscheinlichkeit das Dunkle Materie an einem zufälligem Ort existiert, umgebaut werden. Das NFW-profil (1) wird also zu (3) umgebaut.

$$\rho_{NFWDM}(r) = \frac{1}{\sqrt{2 \cdot \pi} \cdot \sigma} \cdot e^{\left(-\frac{(\Phi(r)}{\sigma^2}\right)} \tag{3}$$

$$\Phi(r) = 1 - \frac{1}{(2 \cdot \sigma^2)} \cdot (M_{xx} \cdot x^2 + 2 \cdot M_{xy} \cdot xy + M_{yy} \cdot y^2)) \tag{4}$$

Eine Mögliche Implementation in der Programiersprache Python als Funktion:

```
def rho(x, y, z):
  a = (1 - ((1) / (2 * (sigma ** 2)))
  b = ( Mxx * x**2 + 2 * Mxy * x * y + Myy * y**2 ) )
  c = a * b
  return rho(x, y, z) * c

def phi(x):
  if x == 0:
    return -4 * pi * f_0 * G * R_s**2

  a = - ( 4 * pi * G * f_0 * R_s ** 3 ) / x
  b = np.log(1. + (x / R_s) )
  c = a * b
  return c
```

2.3 Stauchung und Streckung der Galaxie

Wird eine Galaxie gestreckt oder gestaucht kann das an der umliegenden Dunklen Materie liegen. Um solch eine Streckung darzustellen wird wie folgt vorgegangen: Die Position eines Sternes an einer Achse muss mit einem Skalar multipliziert bzw. dividiert werden. Dies ist relativ einfach machbar da die Koordinaten der jeweiligen Sterne in einer Datei nach dem Format x, y, z gespeichert sind. Um die Galaxie vertikal zu strecken wird z. B. für jeden Stern die z-Koordinate mit dem skalar s multipliziert. Wenn gestaucht werden soll liegt dieser Wert im Intervall $0 < s < 1$. Die neue Koordinate für einen Stern ist also $x, y, z \cdot s$. Möchte man die Galaxie strecken muss das Skalar s im Intervall $1 < s < \infty$ liegen.

Indem man ganz grob feststellt in welchen Bereichen der Galaxie der Anteil an dunkler Materie höher ist kann man dies mit in die Berechnungen einfließen lassen. In meinem Fall habe ich z. B. ausprobiert einen Richtungsvektor $\vec{r}$ zu generieren der von einem Stern in die Richtung der dunklen Materie zeigt. anschließend hab ich den Richtungsvektor mit einem Skalar s multipliziert um die Stärke mit der die Dunkle Energie auf einen jeweiligen Stern wirkt kontrollieren zu können. Als letztes habe ich dann den Richtungsvektor mit mit den Koordinaten des Sternes multipliziert um eine theoretische neue Position für den Stern zu generieren. Tut man dies für alle Sterne entstehen kleinere sogenannte "cluster" in denen sich die Sterne bündeln. Ein Problem hierbei war, dass es unglaublich rechenaufwendig ist dies für mehrere hunderte von Tausenden Sternen zu berechnen (siehe Sektion 2.4 und 2.5)

2.4 Rechenaufwand

Um den Rechenaufwand in der Informatik darzustellen wird die sogenannte "O notation" verwendet. Diese Notation wird verwendet um zu beschreiben wie viele schritte gebraucht werden um an ein Ziel zu kommen, abhängig von der Anzahl der "Objekte":

$$O(n) = |\dots| \tag{5}$$

Beispiel:

$$O(n) = |n^2|$$

Möchte man z. B. die Kräfte zwischen $n = 100$ Sternen berechnen werden $O(100) = 100^2 = 10000$ Rechnungen ausgeführt.

Bei einer "O notation" von n^2 bei der Berechnung von Kräften zwischen den Sternen kann also davon ausgegangen werden das desto mehr Sterne existieren, die Rechenleistung die gebraucht wird um in derselben Zeit dieselbe Anzahl an Sternen zu generieren Exponentiell für jeden Stern Steigen wird.

Eines Optimales Ergebnis wäre eine "Big O notation" von $n \log n$, jedoch ist dies zurzeit nicht ganz möglich.

2.5 Beschleunigung der Generation

Die Geschwindigkeit mit der die Sterne generiert werden ist ohne irgendeine Art von Optimierung unglaublich langsam. Die NFW-Funktion (1) wird für jeden Stern aufs neue vom Computer berechnet was auf mehrere Tausend Sterne hochgerechnet unglaublich rechenaufwendig ist. Ein Weiteres Problem ist, dass das Programm von alleine nur einen Kern verwendet und somit auf eine menge Rechenleistung verzichtet. Durch Nutzung von mehreren Kernen kann die Zeit um n Sterne zu generieren schnell halbiert oder sogar geviertelt werden.

2.5.1 Lookuptable

Eine weitere Möglichkeit für mehrere Berechnungen Zeit zu Sparen ist, den entsprechenden Wert aus dem NFW-Profil (Formel 1) vorher zu berechnen und in eine Tabelle zu schreiben. Dies kann für z. B. $2 \cdot 10^8$ Werte getan werden was zwar eine ca. 6 GB große Datei erzeugt, diese kann jedoch innerhalb weniger Sekunden eingelesen werden und somit das ërrechnenëines entsprechenden Wertes praktisch innerhalb von Bruchteilen einer Sekunde simulieren indem das Ergebnis einfach aus dem Arbeitsspeicher ausgelesen wird.

2.5.2 Mehr Rechenleistung!

Um mehr Sterne in weniger Zeit zu generieren können verschiedene Aspekte der Software optimiert werden. Um jedoch ohne Optimierung mehr Sterne zu generieren kann einfach mehr Rechenleistung verwendet werden. Dies ist im Grunde genommen die einfachste Möglichkeit mehr Sterne in einer relativ kurzen Zeitspanne zu generieren: Schon die Verwendung von vier statt zwei Kernen ermöglicht es einem in 30 Min. statt ca. 300 Sterne ca. 600 Sterne zu generieren.

Amazon Web Services
Um das Generieren von Galaxien so "profitabel"wie möglich zu machen können sogenannte "Amazon Web Services[3] " (AWS) genutzt werden. Der Dienst "EC2" kostet z.B. mit 60 Kernen und 256GB RAM nur $3.712 pro Stunde. Statt mit einem Kern in einer Stunde ca. 600 Sterne zu generieren können also in einer Stunde 38400 Sterne generiert werden! Möchte man $1 \cdot 10^6$ Sterne generieren bräuchte man mit einer Geschwindigkeit von ca. 600 Sternen pro Stunde und 64 Kernen ca. 26 Stunden. Dies kostet umgerechnet ca. 100$ (83€). Eine weitere Möglichkeit besteht darin, Virtuelle Server von Netcup anzumieten. Hierbei kosten z.B. 6 Kerne für einen Monat 8€ wodurch man frei nach Belieben den ganzen Tag Galaxien generieren kann.

2.5.3 Nichts in der Konsole ausgeben

Ein Vorgang der erstaunlicherweise sehr viel Rechenleistung erfordert, ist der Vorgang beim Ausgeben von Text in die Konsole. Gibt man jede potentielle Koordinate in die Konsole aus, stürzt das Programm aufgrund von Zuviel Daten im Arbeitspeicher ab. Um dies zu umgehen kann z. B. nur jeder 100.000 Wert in die Konsole ausgegeben werden was jedoch auch überflüssig ist wenn man ungefähr abschätzen kann, wann das Script fertig gelaufen ist.

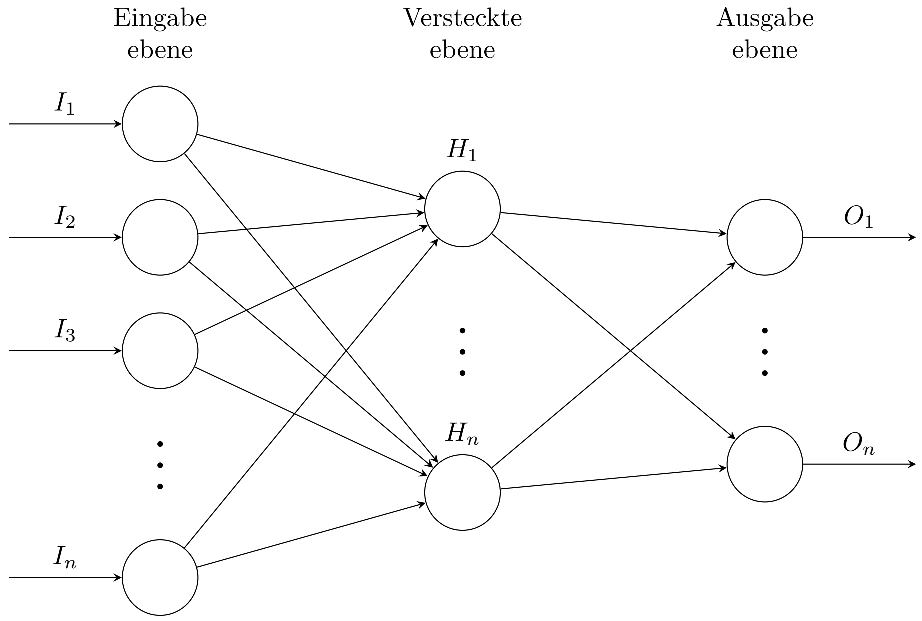
2.6 Nutzung eines neuronalen Netzes zum unbeaufsichtigten generieren von Galaxien

2.6.1 Aufbau des neuronalen Netzes

Ein Neuronales Netz ist wie folgt aufgebaut:

---

[3] Amazon Web Services (AWS) ist ein US-amerikanischer Cloud-Computing-Anbieter, der 2006 als Tochterunternehmen des Online-Versandhändlers Amazon.com gegründet wurde. Zahlreiche populäre Dienste wie beispielsweise Dropbox, Netflix, Foursquare oder Reddit greifen auf die Dienste von Amazon Web Services zurück. 2013 stufte Gartner AWS als führenden internationalen Anbieter im Cloud Computing ein.



Im Grunde genommen werden Daten eingespeist und miteinander verrechnet, wodurch am Ende ein oder mehrere Werte rauskommen mit denen man die verschiedensten Sachen tun kann. In meinem Fall konnte ich z. B. die durchschnittliche Dichte von Sternen, die Größe der Galaxie und viele andere Faktoren in das neuronale Netz einspeisen um am Ende zwei Werte entnehmen. Das neuronale Netz muss trainiert werden, dabei werden echte funktionierende Daten in das Netz eingespeist und mit bereits vorhandenen Ergebnissen verglichen. Ist das Ergebnis gut und stimmt ungefähr mit dem bereits vorhandenen Ergebnis überein wird am Netz selber nichts getan. Stimmt das Ergebnis aus dem Netz mit dem bereits vorhandenen jedoch nicht überein, dann werden im neuronalen Netz die sogenannten Synapsen (Die Verbindungen zwischen den Neuronen (Oben als Kreis dargestellt)) entsprechend anders gewichtet.

Neuronen und Synapsen In einem neuronalen Netz sind sogenannte Neuronen (In der Abbildung oben als Kreis dargestellt) über sogenannte Synapsen (In der Abbildung oben als Linie zwischen zwei Neuronen dargestellt) miteinander verbunden. Die Neuronen können als eine Art Funktion gesehen werden. Sie Wandeln die Daten die sie bekommen mithilfe einer Aktivierungsfunktion in einen Zahlenbereich zwischen 0 und 1 um. Eine solche Aktivierungsfunktion kann die folgende Form haben:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

Die Synapsen können verschieden gewichtet sein. Bekommt eine Synapse z. B. einen Eingabewert x, dann kann der Eingabewert mit der Gewichtung w der Synapse verrechnet werden um den Ausgabewert a zu erhalten:

$$x \cdot w = a \tag{7}$$

Für eine weite Ausführung in den Bereich der neuronalen Netze reicht die vorgegebene Maximalanzahl an Seiten leider nicht, deshalb hier kurz das wesentliche: Um neuronale Netze effektiv nutzen zu können wird unglaublich viel Rechenleistung benötigt. Diese habe ich nicht einfach so zur Verfügung, weshalb ich Kontakt mit verschiedenen Unis und Unternehmen aufgenommen haben um dort vielleicht Zugang zu einem Hochleistungsrechner zu bekommen. Zum Zeitpunkt der Abgabe dieser Langfassung (9. Februar 2018) habe ich jedoch noch keine Möglichkeit gehabt meine Software auf einem solchen Hochleistungs-Rechner laufen zu lassen. Deshalb habe ich beschlossen die Nutzung von neuronalen Netzen nach hinten zu verschieben, auch wenn das Thema unglaublich spannend ist.

2.7 Spiralgalaxien

Spiralgalaxien sind im allgemeinen faszinierende Gebilde: Aus mehreren Millionen Sternen entsteht eine Reisige Spirale. Dies zu simulieren ist jedoch unglaublich Rechenaufwendig weshalb ich dies bisher nur mir kleineren Galaxien durchgeführt habe. Das Problem ist, dass die Kräfte zwischen jedem Stern und jedem anderem Stern ausgerechnet werden müssen was wie in Sektion 2.4 beschrieben mit steigender Anzahl an Sternen eine Exponentielle Steigerung der Rechenzeit hervorruft.

Ein interessanter Aspekt der Spiralgalaxien den ich in die Simulationen einzubauen ist die Diskrepanz zwischen der realen Position der Sterne und der berechneten Position durch die Auf Dunkle Materie geschlossen wird.

2.7.1 Das n-Körper Problem

(nach der Ph.D. Thesis von Jakub Schwarzmeier s. 18 - 22, Pilsen 2007)

Das sogenannte N-Körper Problem wird dazu genutzt um ein System mit N-Körpern zu simulieren. Hierbei müssen alle von außen einwirkenden Kräfte $\vec{F_i}$ mit eingerechnet werden, im falle von Galaxien also die universelle Gravitationsregel von Newton.

$$m_i \cdot \frac{d\vec{v_i}}{dt} = \vec{F_i} \tag{8}$$

Für zwei Punkte in einer Galaxie gilt nach der universellen Gravitationsregel von Isaac Newton:

$$m_i \cdot \frac{d\vec{v_i}}{dt} = G \cdot \frac{m_i \cdot m_j}{r_{ij}^3} \cdot \vec{r_{ij}} \tag{9}$$

bzw.

$$\frac{d^2\vec{r_i}}{dt^2} = G \cdot \sum_{j=1} \frac{m_j}{r_i^3 j} \cdot \vec{r_{ij}} \tag{10}$$

Die neue Position und Geschwindigkeit eines Körpers wird mithilfe der bereits bekannten Beschleunigung berechnet, was zu der Formel (10), die eine zweite Ableitung enthält, führt. Die Formel (10) kann jedoch in zwei neue Formeln umgeschrieben werden, die jeweils nur eine erste Ableitung enthalten:

$$\frac{d\vec{r_i}}{dt} = \vec{v_i} \tag{11}$$

$$\frac{d\vec{v_i}}{dt} = G \cdot \sum_{j=1} \frac{m_j}{r_{ij}^3} \cdot \vec{r_{ij}} \tag{12}$$

Die Formeln (11) entsprechen der Bewegungsgleichung nach Hamilton. Möchte man nun die Position der einzelnen Sterne berechnen, müssen die Werte aus der Funktion in für einen Computer darstellbare Werte umgewandelt werden.

Eines der Probleme um die Bewegung eines Sternes zu formulieren liegt dabei, einen Anfangswert für die Bewegung zu finden. Dies wird durch die folgende Formel gelöst:

$$x_{t+1} = x_t + \Delta t \cdot F(x_i, t_i) \tag{13}$$

Die Position eines Sternes nach einer Zeit von Δt wird durch die vorherige Position und die auf der Stern wirkenden Kräfte bestimmt. Der Term $\Delta t \cdot F(x_i, t_i)$ wird auch als erster Schritt in der Taylor-reihe bezeichnet mit der weitere Punkte anhand der Ableitung vorheriger Punkte errechnet werden können. Die Veränderung der Position kann aus der Formel (11) abgeleitet werden:

$$\Delta \vec{r_i} = \vec{v_i} \cdot \Delta t \tag{14}$$

$$\Delta \vec{v_i} = G \cdot \Delta t \cdot \sum_{j=1} \frac{m_j}{r_{ij}^3} \vec{r_{ij}} \tag{15}$$

2.7.2 Unterteilung des Vektorraumes in verschiedene Zellen

Die Kräfte die innerhalb der Galaxie wirken können mithilfe eines Vektorraumes dargestellt werden. Dabei kann jedem Punkt im Raum ein Vektor zugewiesen werden. Der Vektorraum im falle von Galaxien stellt erstmal nur die Kräfte, die die Sterne aufeinander auswirken da. Um nicht unendlich viel rechnen zu müssen wird der Vektorraum in verschiedene Zellen unterteilt in denen jeweils die mittlere Kraft berechnet wird.

2.7.3 Berechnung der wirkenden Kräfte

Die wirkenden Kräfte können wie in Formel (16) zu sehen berechnet werden:

$$F_1 = F_2 = G \cdot \frac{m_1 \cdot m_2}{\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2}} \tag{16}$$

F_1, F_2	Wirkende Kräfte zwischen zwei Massen
G	Gravitationskonstante $6,67408 \cdot 10^{-11} \frac{m^3}{kg \cdot s^2}$
m_1	Erste Masse
m_2	Zweite Masse
r	Abstand der Massen

Masse der Sterne Die Masse der Sterne ist einer der entscheidende Faktoren wenn es darum geht Galaxien zu generieren: verändert man die Masse der Sterne verändert sich sofort das gesamte Gleichgewicht in der Galaxie was zu unerwarteten Ereignissen führen kann.

Allgemein gesehen werden zwei Variablen gebraucht: die **Minimalmasse** und die **Maximalmasse**. Zwischen diesen beiden Werten werden zufällig Werte generiert und den Sternen zugewiesen.

Die Veränderung der Masse wird erstmal nicht berücksichtigt.

Abstand der Sterne Der Abstand der Sterne kann mit dem Satz des Pythagoras (Formel (17)) berechnet werden.

$$r_{a,b} = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_z - b_z)^2)} \tag{17}$$

2.8 Weiteres

Um die Simulation zu beschleunigen können andere Simulationsmodelle verwendet werden wie z. B. die Kräfte in verschiedenen Feldern berechnen um diese anschließend weiter zu evaluieren.

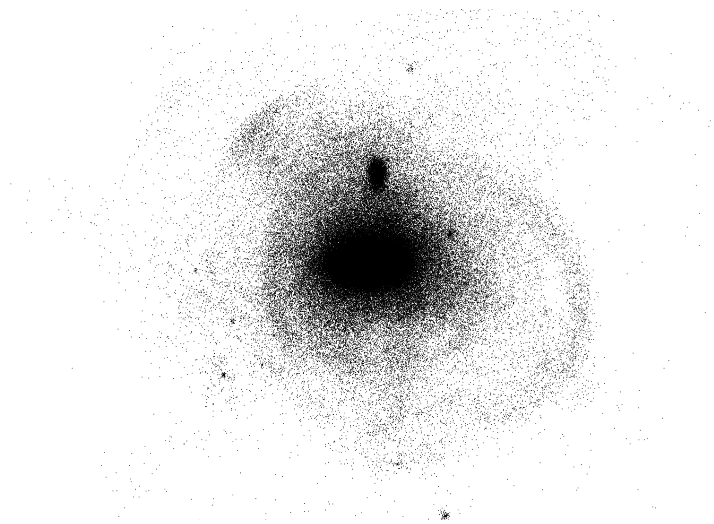


Abbildung 4: Eine Spiralgalaxie generiert mithilfe von Daten aus dem Max-Plank-Institut in Heidelberg

3 Ergebnisse

3.1 Simulations Geschwindigkeit

Insgesamt bin ich zu folgenden Ergebnissen zusammengekommen:

Sterne	Zeit (30.10.2017)	Zeit (9. Februar 2018)
45000	ca. 9h	ca. 4h

- Pro MegaByte können die Koordinaten von 10000 Sternen gespeichert werden.
- Die Nutzung von Lookuptabellen ist unglaublich sinnvoll

3.2 Lookuptabellen Geschwindigkeit

Rho-werte	Schrittweite	Zeit zum einlesen (in sekunden)
1500000	1	8.07
750000	2	4.4
375000	4	2.26
187500	8	1.35
93750	16	0.76

Hier ist klar zu sehen, dass es eine lineare korrelation zwischen der Anzahl an generierten Werten und der Zeit gibt.

3.3 Fazit

Insgesamt betrachtet kann ich behaupten das das Projekt ein voller Erfolg war: Ich habe unglaublich viele neue sachen gelernt und dabei ein Funktionierendes Program zur visualisierung von Galaxien und DUnkler Materie Gebaut. Dabei bekam ich einblicke in die verschiedensten Teilgebiete der Physik, AstroPhysik, Matematik und Informatik. Viele dieser Themen konnte ich jedoch aufgrund ihrer Komplexität nur in geringen maßen nutzen, weshalb ich mich in der Zukunft gerne damit auseinander setzen würde. Ein

Beispiel hierfür sind die Neuronalen Netze welche ein enormes Potential haben welches ich unbedingt nutzen möchte.

4 Quellen und Hilfen

Das Python-Programm sowie die Blender Darstellungen wurden vollständig ohne fremde Hilfe selber erstellt.

Einen Großteil der Formeln fand ich durch eine Wikipedia Recherche.

Das Programieren in der Programiersprache Python habe ich während meines Jugen-Forscht Projektes im letztem Jahr (2017) gelernt. Mit dem Umgang des 3D-Programms Blender bin ich schon vertraut gewesen. Die Grundlagen für LaTeX, in dem diese Langfassung geschrieben wurde, erlernte ich durch das Studieren diverser Beiträge in Foren und der Einsicht in das Jugend Forscht Projekt von Konstantin Bosbach, Tilman Hoffbauer und Steffen Ritsche (2016, Underwater Accoustic Communication). Die Einführung in die Mathematik bekam ich während meines Praktikums im Zentrum Für Astronomie in Heidelberg durch Tim Tugendhat.

Dank gilt...

Herrn Jörg Thar meinem Betreuer

Tim Tugendhat der mir es ermöglichte ein Praktikum im Astronomischen Recheninstitut zu machen.

Konstantin Bosbach welcher mir eine Möglichkeit gab für 2 Wochen in Heidelberg zu wohnen.

Tilman Hoffbauer der bei Problemen bereit war Licht ins Dunkle zu bringen.

Außerdem gilt mein Dank allen, die mich auf jede nur erdenkliche Weise unterstützt haben.

5 Nach der Abgabe...

5.1 Spiralgalaxien

5.2 Using Object Oriented Programming (OOP) techniques

In my case, the objects are galaxies.

5.2.1 Initialisation

The galaxy is initialised with the following objects:

- A list storing the coordinates of each star
  - X Coordinate
  - Y Coordinate
  - Z Coordinate
- A list storing the individual forces acting on the stars
  - X Force
  - Y Force
  - Z Force
- A variable storing the number of stars generated in the galaxy
- Newtons gravitational constant

5.3 Generation of new stars

The function is given an integer defining the amount of stars that should be newly generated.

The newly generated stars are then appended to the list storing the coordinates.

The counter counting the amount of stars in the galaxy is incremented.

5.4 Printing all the coordinates

The function cycles through the list storing the star coordinates and prints them to the command line.

5.5 Calculating the Forces acting between the Stars

The function recieves two star objects and an axis on which the forces should be calculated and returns the force acting on the given axis. In case of a failture (The two given stars have got the same coordiantes), the function just goes on to the next Star.

The Forces can be calculated using the equation (16).

5.6 Calculating the forces acting between each star in the galaxy and each other star

To calculate the forces inbetween every star in the galaxy, the function cycles through every star, looks if the force that should be calculated hat not been calculated yet and calculates it. This includes testing if the force that should be calculated is not the force inbetween a star and itself.

The results of the calculations are stored in a list storinf the forces.

5.7 Printing all the individual forces

The function is able to print all the forces acting inbetween the stars if no argument is given. If an argument n is given, the function print out the nth star in the list.

5.8 Spherical cells

5.8.1 Testing if a point is inside or outside a sphere

In order to test is a point is inside a sphere, one just has to test if the following conditions are all true:

$$\begin{aligned} S_x - S_r \leq P_x \leq S_x + S_r \\ S_y - S_r \leq P_y \leq S_y + S_r \\ S_z - S_r \leq P_z \leq S_z + S_r \end{aligned} \tag{18}$$

P_x , P_y and P_z are the coordinates of the point to be tested, S_x , S_y and S_z are the coordinates of the midpoint of the sphere and S_r is the radius of the sphere.

5.8.2 Testing if a star is inside or outside of a sphere for a whole galaxy

While testting if a star is inside a sphere or not, because of the alignment of the spheres, a point can be in more than one sphere at the same time. To get rid of this problem, the software cycles over every star and searches for matches within the spheres. If a match is found, the next star is tested. This is pretty much as efficient as it can get.

$$O(n) = n_{stars} \cdot n_{spheres} \tag{19}$$

5.8.3 Generate the position of the spheres

Generating the position of the spheres is accomplished in the following way: A 3D-grid is generated and the midpoints of the spheres are positioned on the gridpoints.

[Include Graphic]

The distance the spheres have to each other ist defined using the following function:

$$\texttt{sphere\_distance} = \frac{\texttt{galaxy\_range}}{\texttt{sampling\_rate}} \tag{20}$$

The higher the `sampling_rate` is, the more spheres get generated.

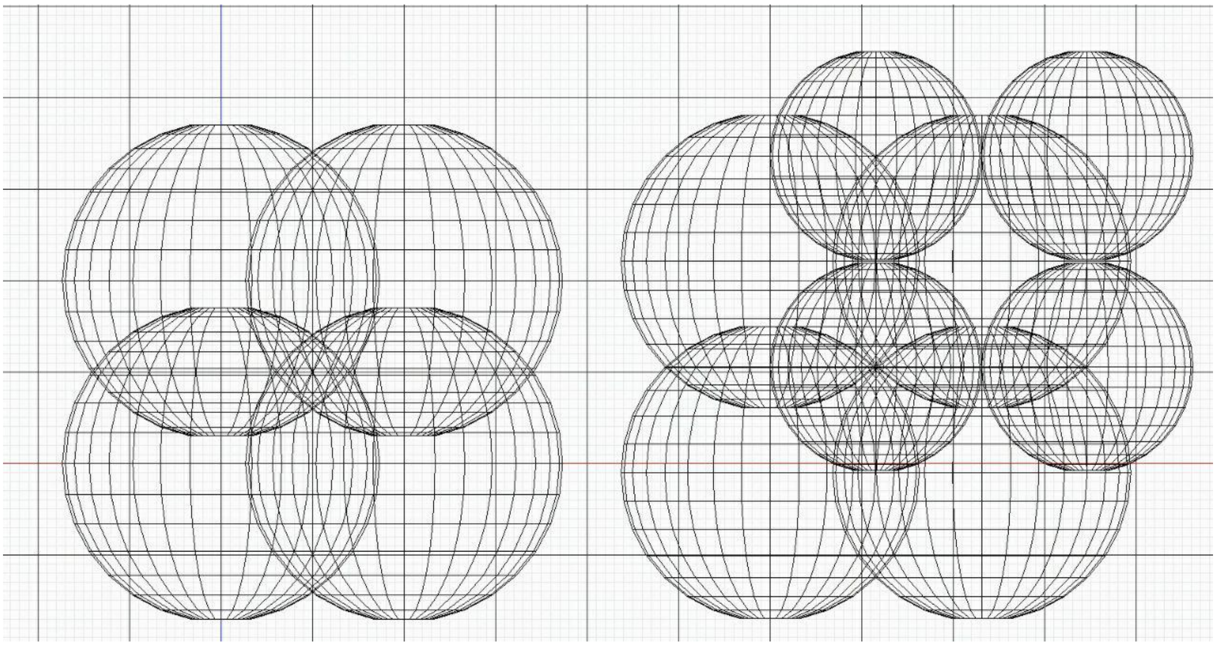


Abbildung 5: The Alignment of the spheres
Left: The perfect alignment covering the complete space
Right: A previously generated alignment using small spheres to cover the missing space

The next goal is to find out the "sweet spot" generating the spheres. When using a very low `sampling_rate`, the reault gets inacurate, but when using a high `sampling_rate`, the calculations are not affected in term of speed and efficiency. The Goal is therefor to find a sampling rate that enables the generation of accurate but fast results.

By being able to controll the accuracy anf therefor the time, it is possible to teach the system to generate a galaxy in like one hour and it will automatically set the sampling rate so low that the simulation will finish perfectly in time.

5.8.4 The radius of the spheres

The Radius of the spheres is dynamicly allocated to ensure that the whole galaxy is covered.

$$r = \sqrt{\texttt{sphere\_distance}^2 + \texttt{sphere\_distance}^2 + \texttt{sphere\_distance}^2} \tag{21}$$

$$1.7320508075688772 = \sqrt{1^2 + 1^2 + 1^2}$$

The equation (21) highly depends on the equation (20) and it's parameters.

5.8.5 Calculate the forces acting on the spheres

In order to reduce the time that is needed to calculate the forces inbetween the stars, the stars are subdivided in different cells, in this case spheres. After all the forces acing inside one sphere are calculated, the forces are combined and applied to the midpoint of the sphere genrating a new coordinate: the mean force. The mean force inbetween all the cells can be calculated.

[Include description binary tree]

[Include graphic binary tree]

5.8.6 Calculate the forces acting on all the spheres together

This should be 0.

5.8.7 Benchmarks

Nr of Stars	Sample rate	Galaxy Range	Time (s)
100	1	100	0.0814
75	1	100	0.0499
50	1	100	0.0295
25	1	100	0.0116
100	1	100	0.0828
100	2	100	0.0909
100	4	100	0.1832
100	8	100	1.1114
100	16	100	7.6944
100	32	100	56.5731
100	64	100	217.7768
100	1	1	0.0809
100	1	2	0.0844
100	1	4	0.0782
100	1	8	0.0758
100	1	16	0.0847
100	1	32	0.0815
100	1	64	0.0770

The sample rate is the factor that influences the time the most. Knowing this, it (the sample rate) can be used to controll the time in which a galaxy can be created. This is usefull in particular when using some very powerfull mashine with limited time.

5.9 Calculate the Position of a Star after a timestep

Not to be considered:

- drag
- any kind of resistance
- acceleration