

Galaxy Generation

Jugend Forscht 2017

Emile Hansmaennel `emile.hansmaennel@gmail.com`

12. Dezember 2017

Ziel meines Projektes war es, Galaxien zu generieren und diese mit den Modellen von echten Galaxien zu vergleichen um mit der Zeit immer bessere Simulationen zu generieren. Dies gelang mir im Verlauf des letzten Jahres immer besser, wobei ich zurzeit (12. Dezember 2017) auf meinem Laptop ca. 1000 neue Sterne in 1.5 Stunden generieren kann. Zu dem Projekt bin ich während meines Praktikums im Zentrum für Astronomie in Heidelberg (ZAH) gekommen bei dem ich von Tim Tugendhat eine Einführung in das NFW-Profil bekommen habe, welches genutzt wird um Galaxien zu generieren.

Inhaltsverzeichnis

1	Einleitung	2
1.1	Themen	2
1.2	Motivation	2
2	Hauptteil	3
2.1	Generierung der Elliptischen Galaxien	3
2.1.1	Das Navarro-Frenk-White Profil	3
2.1.2	Random Sampling	3
2.2	Generierung eines Dunkle-Materie Halos	3
2.2.1	Anpassung des NFW-Profiles	4
2.3	Stauchung und Streckung der Galaxie	4
2.4	Beschleunigung der Generierung	5
2.4.1	n-Sterne	5
2.4.2	Lookuptable	5
2.4.3	Weitere Optimierungen	5
2.5	Nutzung eines Neuronalen Netzes zum unbeaufsichtigeten generieren	6
2.5.1	Aufbau des Neuronalen Netzes	6
2.5.2	Nutzung eines Neuronalen Netzes zur verbesserung von Galaxien Simulationen . .	6
2.6	Spiralgalaxien	7
2.6.1	Das n-Körper Problem	7
2.7	Größeneinheiten	7
3	Ergebnisse	8
3.1	Simulation Speed	8
3.2	Spiral Galaxies	8
3.3	Lookup-Table Speed	8
3.4	Distortion of Galaxies	9
4	Quellen und Hilfen	10

1 Einleitung

Nach meinem letzten Jugend-Forscht Projekt ergab sich die Möglichkeit ein Praktikum im Zentrum für Astronomie in Heidelberg zu absolvieren. Über die social-media Plattform Reddit stellte ich den kontakt mit Tim Tugendkat 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 erlernte Fähigkeit mit Python¹ zu Programmieren und mit Blender² 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

Ich habs einfach mal getan...

¹Programmiersprache

²3D Software Suite

2 Hauptteil

2.1 Generierung der Elliptischen Galaxien

2.1.1 Das Navarro-Frenk-White Profil

Das Navarro-Frenk-White profil (NFW-profil) ist im grunde genommen eine Funktion die einem die Wahrscheinlichkeit das ein Stern an einer bestimmten position ist liefert. Die Funktion ist im allgemeinen wie folgt aufgebaut:

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

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

Um die Formel (1) einfach zu beschreiben kann man sie sich wie folgt vorstellen: Um zu gucken ob ein zufälliger Stern bei x_1 , y_1 und z_1 generiert werden kann wird wie folgt vorgegangen: Aus den Koordinaten wird der Wert r mithilfe des Satz des Pythagoras berechnet ($r = \sqrt{x_1^2 + x_2^2 + x_3^2}$), dieser gibt an wie weit der jeweilige Stern vom Zentrum der Galaxie entfernt ist. Um zu prüfen ob der Stern generiert wird, wird dieser r -wert in die Funktion ρ_{NFW} eingesetzt. Der entstehende Wert gibt an wie wahrscheinlich es ist, das ein Stern in der Entfernung zum Ursprung generiert wird.

2.1.2 Random Sampling

Die sogenannte "Random Sampling" Methode wird genutzt um herauszufinden ob ein Stern generiert wird oder nicht. Es wird dazu ein zufälliger Wert x im bereich $[\rho_{max} ; \rho_{min}]$ generiert. Liegt dieser Wert über dem Wert aus der Funktion ρ wird kein Stern generiert. Liegt dieser Stern jedoch unter dem wert aus der ρ Funktion wird ein Stern an den Koordinaten x_1 , y_1 und z_1 generiert.

Um das generieren zu Beschleunigen wird eine sogenannte "lookuptable" verwendet. (→ 2.4.2)

Generiert man ein paar Sterne mithilfe des NFW-Profiles hat man theoretisch schon eine Galaxie, jedoch ist diese nicht klar definiert. Um eine klare definition zu erreichen müssen mehrere hundert Sterne generiert werden.

2.2 Generierung eines Dunkle-Materie Halos

Das sogenannte "Dunkle-Materie Halo" ist eine art Kugel die eine Galaxie umspannt: Durch dieses Halo ist die Dichte der Dunklen Materie welches sich um die Galaxie herum befindet definiert. Problematisch ist jedoch, dass wir dieses Halo nicht sehen können weshalb wir nur aufgrund anderer phänomäne welche

durch die Halos verursacht werden auf die Eigenschaften des Halos schließen können.

Um diese Halos darzustellen wird das NFW-Profil (1) abgewandelt und quasi mit dem Profil für Elliptische Galaxien verbunden.

...

2.2.1 Anpassung des NFW-Profiles

$$\rho(r) = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma}} \cdot e^{\left(-\frac{\Phi(r)}{\sigma^2}\right)} \quad (2)$$

$$\rho(r) \cdot 1 - \frac{1}{(2 \cdot \sigma^2)} \cdot (M_{xx} \cdot x^2 + 2 \cdot M_{xy} \cdot xy + M_{yy} \cdot y^2) \quad (3)$$

```
1
2 # new rho function
3 def rho_new(x, y, z):
4     a = (1 - ((1) / (2 * (sigma ** 2))))
5     b = ( Mxx * x**2 + 2 * Mxy * x * y + Myy * y**2 )
6     c = a * b
7     return rho(x, y, z) * c
8
9 # phi function
10 def phi(x):
11     if x == 0:
12         return -4 * pi * f_0 * G * R_s**2
13
14     a = - ( 4 * pi * G * f_0 * R_s ** 3 ) / x
15     b = np.log(1. + (x / R_s) )
16     c = a * b
17     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. Da 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.

2.4 Beschleunigung der Generierung

Die Sterne schnell zu generieren ist natürlich energieeffizienter aber auch wichtig damit das neuronale netzt in unserer lebzeit fertig wird.

Es gibt ein paar Aktionen die umgebaut werden können um das generieren zu beschleunigen:

2.4.1 n-Sterne

Statt am Anfang mehrere Millionen Sterne zu generieren wird wenn eine neue Koordinate benötigt wird eine neue erstellt. So erstellt man auf keinen Fall zu viele Koordinaten was Zeit spaart. Dem programm kann also gesagt werden, dass es genau n_1 Sterne aus m_1 potentiellen Sternen generieren soll, andernfalls werden n_2 Sterne aus $m_2 > m_1$ potentiellen Sternen generiert.

2.4.2 Lookuptable

Eine Weitere Möglichkeit für mehrere Berechnungen Zeit zu Spaaren ist, den Entsprechenden Wert aus dem NFW-Profil (Formel 1) vorher zu berechnen und in eine Tabelle zu schreiben. Dies kann für z.B. $2e8$ Werte getan werden was zwar eine 6 GB große Datei erzeugt, diese kann jedoch innerhalb weniger Sekunden eingelesen werden.

2.4.3 Weitere Optimierungen

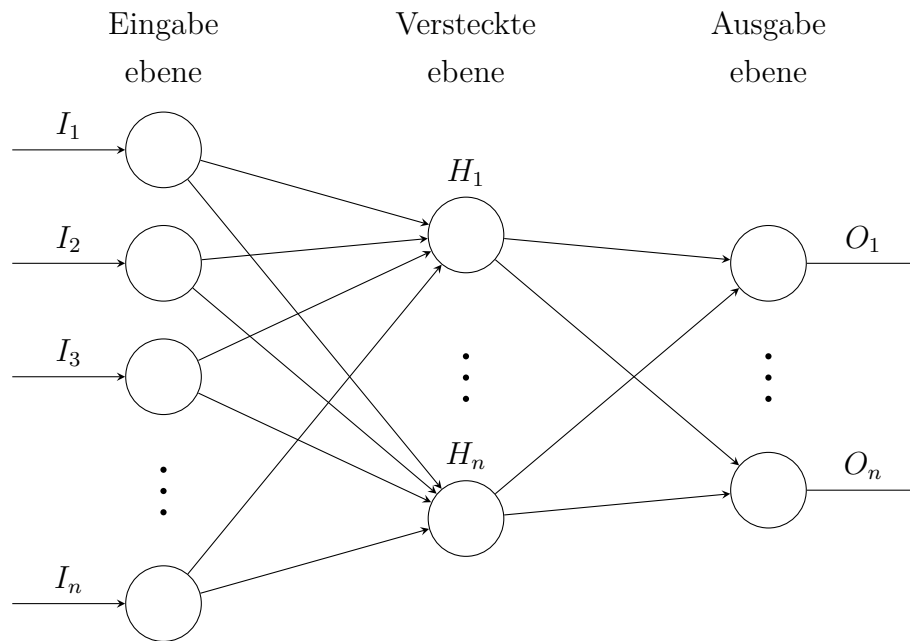
Nichts in der Konsole ausgeben: Eine Vorgang der erstaunlicherweise sehr viel Rechenleistung erfordert, ist der Vorgang beim ausgeben von Text in die Konsole. Gibt man jede potentielle Koordinate in die Konsole aus, stürzt das Programm aufgrund von Überlast ab. Um dies zu umgehen kann z.B. nur jeder 100.000 Wert in die Konsole ausgegeben werden.

...

2.5 Nutzung eines Neuronalen Netzes zum unbeaufsichtigten generieren

2.5.1 Aufbau des Neuronalen Netzes

Ein Neuronales Netz ist wie folgt aufgebaut:

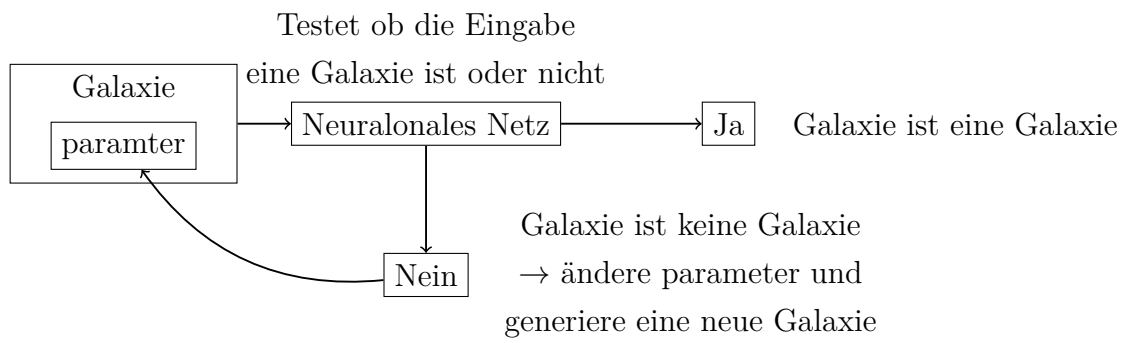


Das **Neuronale Netz** besitzt mehrere Ebenen: die **Eingabe ebene**, die **Versteckte Ebene(n)** und die **Ausgabe Ebene**. Diese Ebenen bestehen aus sogenannten **Neuronen**, die wie im menschlichen Gehirn Informationen aufnehmen und weitergeben. Die Eingabe kann verschieden gewichtet sein, es kann also sein, dass eine Eingabe eine Gewichtung von 10% hat und eine andere eine Gewichtung von 90%. Die Eingabe Ebene ist dazu da, eine Eingabe in einer Matrix an die verschiedenen Neuronen in der Versteckten Ebene weiterzuleiten. Die Versteckte Ebene verarbeitet die Information aus der Matrix und leitet diese an die Ausgabe Ebene weiter, die die Information ausgibt.

Das sogenannte "Trainieren" ist der Prozess, bei dem die Gewichtung der Neuronen so verändert wird, dass ein gewünschtes Ergebnis herauskommt. Beispiel: man möchte ein neuronales Netz darauf trainieren, eine Galaxie zu identifizieren. Dann werden ganz viele positive Beispiele durch das Netz gejagt, welche die Gewichtung immer weiter anpassen. In der Ausgangs Ebene wird dann mithilfe zweier Neuronen entweder dargestellt, ob das eingegebene Bild eine Galaxie ist oder ob das eingegebene Bild eben keine Galaxie ist.

2.5.2 Nutzung eines Neuronalen Netzes zur Verbesserung von Galaxien Simulationen

Möchte man mithilfe eines neuronalen Netzes vorhandene Galaxiensimulationen verbessern, wird wie im folgenden Diagramm zu sehen vorgegangen:



2.6 Spiralgalaxien

2.6.1 Das n-Körper Problem

2.7 Größeneinheiten

$$3.086 \cdot 10^{36} m \quad (4)$$

3 Ergebnisse

Ergebnisse

3.1 Simulation Speed

Nach mergen des speed-branches sind folgende Ergebnisse zusammengekommen:

Sterne	Zeit Vorher	Zeit Nachher
1e5	2.93 sek.	k.A.
1e6	29.38 sek.	k.A.
1e7	315.67 sek.	k.A.
1e9	9h	k.A.

Aus 1e9 Sternen werden vorher letztendlich 45000 Sterne generiert.

Pro MegaByte können die Koordinaten von 10000 Sternen gespeichert werden.

3.2 Spiral Galaxies

Die generierung von Spiralgalaxien gestaltet sich schweiriger als erwartet.

3.3 Lookup-Table Speed

rho-values	step	time (in seconds)
1500000	1	8.07
750000	2	4.4
375000	4	2.26
187500	8	1.35
93750	16	0.76

The correlation between the number of stars generated and the time needed ist clearly linear.

Python script

```
1 import matplotlib.pyplot as plt
2
3 list_time = [8.07, 4.4, 2.26, 1.35, 0.76]
4 list_rho_values = [1500000, 750000, 375000, 187500, 93750]
5
6 plt.plot(list_time, list_rho_values, '-ro')
7 plt.show()
```

3.4 Distortion of Galaxies

Galaxien verformen dinge

4 Quellen und Hilfen

Quellen

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).

Dank gilt...

Herrn Jörg Thar meinem Betreuer

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.