Galaxy Simulation

Jugend Forscht 2018/2019

Emile Hansmaennel

2018 - 2019

Zusammenfassung

Ist es möglich die Entstehung von Galaxien zu simulieren? Um diese Frage zu beantworten bin ich zu dem Schluss gekommen, dass ich das doch einfach mal ausprobieren sollte. Dazu habe ich das Navarro-Frenk-White Profil implementiert um anschließen die Kräfte die Zwischen den Sternen wirken zu berechnen. Dabei stattete ich die Sterne mit einer zufälligen Masse aus und Unterteilte die Galaxie in dynamisch-große Zellen um die Simulation stark zu beschleunigen. Um eine Stabile Galaxie zu simulieren müssen jedoch alle Sterne in der Galaxie eine Anfangsgeschwindigkeit besitzen die sie auf eine Kreisbahn lenkt, damit die Kraft, welche die Sterne in die Mitte der Galaxie zieht ausgeglichen werden.

Inhaltsverzeichnis

1 Einleitung

Das Projekt ist nach meinem vorletzten Jugend-Forscht Projekt entstanden: Ich habe ein Praktikum in Astronomischen Recheninstitut in Heidelberg genutzt, um mit einem Doktoranden[1] das Navarro-Frenk-White Profil, das zum generien von Punktwolken genutzt wird, zu visualizieren. Anschließend hat sich das Projekt ein bisschen verlaufen, irgendwann beschloss ich jedoch, dass das Projekt weiterzuführen und statt nur statische Galaxien zu generieren dazu überzugehen die Galaxien zu simulieren, also die Entwicklung einer virtuellen Galaxie zu untersuchen. Eines der Entscheidenden Probleme war die Laufzeit der Simulation. Das Problem das es zu lösen galt, war die nutzung von mehreren Threads mit der nutzung des Barnes-Hut Algorithmus zu kombinieren. Das Ergebniss ist sehr schön: Durch die nutzung der Programmiersprache Golang war das einbauen der nutzung von mehreren Threads vergleichsweise einfach.

2 Vorgehensweise

Wie schon in der Einleitung beschrieben habe ich mehrere Techniken kombiniert um mein Ziel zu erreichen. Das komplette Projekt lässt sich in mehrere Abschnitte unterteilen: Die Generierung der Punktwolke, aus der eine Galaxie abstrahiert wird. Die Simulation der Galaxie bei der die Kräfte die in der Galaxie wirken berechnet werden und daraus die Geschwindigkeit und Richtung in die der Stern fliegt.

3 Generieren

Das Generieren der Statischen Punktwolke aus der die Galaxie abstrahiert wird ist ein wichtiger Bestandteil des Gesamtprojektes, denn alles baut auf ihr auf. Kurz: um

[1] Tim Tugendhat

Kräfte zwischen Sternen zu berechnen braucht man erstmal Sterne!

3.1 Das Navarro-Frenk-White Profil

Das Navarro-Frenk-White Profil (NFW-profil) ist ein Profil zur Generierung von Koordinaten in n-Dimensionalen Systemen. Das Profil gibt einem die Warscheinlichkeit ρ zurück, dass ein Punkt im Abstand r zum Mittelpunkt des raumes existiert. Die dazugehörige Funktion sieht wiefolgt aus:

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

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

Möchte man nun herausfinden wie weit ein Punkt mit der Koordinate $(x_1, x_2, ..., x_n)$ mit $x \in \mathbb{N}$ vom Mittelpunkt des Raumes entfernt ist, kann der Satz des Pythagoras (2) verwendet werden.

$$r_n = \sqrt{\sum_{i=0}^{n} x_i^2} \qquad n \in \mathbb{N} \tag{2}$$

Der Abstand r zum Mittelpunkt des Raumes kann nun in das NFW-Profil 1 gegeben werden wodurch ein Wert s entstehet:

$$\rho_{NFW}(r) = \cdots = s \tag{3}$$

Dieser Wert s stellt die Warscheinlichkeit dar, das ein Stern der eine Entfernung r vom Mittelpunkt der Galaxie besitzt existiert.

Die Galaxie Wirkt nun aus der Ferne wie ein Würfel, da die aus ρ retultierende Kurve aprubt endet. Dies kann gelöst werden, indem statt $\rho_{NFW}(r)$ folgendes gerechnet wird: $\rho_{NFW}(r) - \rho_{NFW}(r_{max})$

3.2 Random Sampling

Sei s ein zufälliger Wert im Intervall $\psi = [\rho(r_{min}); \rho(r_{max})]$. Generiert man nun einen zufälligen Wert r im Intervall ψ, kann man schauen ob $s > r \vee s < r$ gilt. Ist $r > s$, wird der Stern verworfen, ist $r < s$ wird der Stern behalten.

3.3 Lookup Tabellen

Statt bei der generierung eines Punktes jedes mal das NFW-Profil (1) anzuwenden, kann das NFW-Profil für einen Bereich vorberechnet werden und in einer Tabelle abgespeichert werden. Somit kann wenn eine Entfernung r zum Mittelpunkt des Raumes vorliegt der entsprechende Wert aus der Tabelle ausgelesen werden. Die Tabelle kann jedoch nicht so genaue Ergebnisse liefern wie das NFW-Profil, sie kann jedoch so angepasst werden, dass sie in den Arbeisspeicher passt und somit das Generieren stark verschnellert. Mit genügend Arbeitsspeicher ist der Fehler auch vernachlässigbar.

3.4 Beschleunigung der Generierung

Es existieren mehere Möglichkeiten das Generierung der Punkte zu verschnellern.

Eine gute Möglichkeit ist die Nutzung von mehr Rechenleistung. Bei der Nutzung von n Rechenkernen ist das Generieren von Sternen n mal schneller. Die Server des Server-Hosters Hetzner können dabei gut verwendet werden: Es wird stündlich aberechnet und 32 Kerne mit 128 GB RAM kosten $\approx$ 50ct / h was es ermöglicht für einen vergleichsweisen Günstigen Preis, sehr viele Koordinates zu generieren.

Die Ausgabe von jeder Potentiellen Koordinate in die Kommandozeile verlangsamt die Generierung unglaublich stark, da der Rechner darauf wartet das die Ausgabe fertig ist bevor er mit der nächsten rechnung beginnt was zu einer relativ starken verlangsamung der Generierung führt.

4 Simulieren

4.1 Die Entstehung von Galaxien

“Eine Galaxie ist eine durch gravitation gebundene große Ansammlung von Sternen, Planetensystemen, Gasnebeln und sonstigen Stellaren Objekten.“ [2]

Demnach ist es relativ Einfach eine Galaxie zu generieren: es werden einfach ganz viele Objekte in einen Raum geworfen. Das reicht jedoch nicht um die Objekte als Galaxie definieren zu können, da sie nicht “durch Gravitation gebunden“ sind.

Um dies zu tun muss die Kraft zwischen allen Objekten in der Galaxie berechnet werden und damit die Position des Objektes nach einer bestimmten Zeit bestimmt werden.

Dies reicht jedoch auch nicht um eine “stabile“ Galaxie zu generieren: berechnet man nur die Kräfte die auf ruhende Objete in einem Reibungfreiem Raum wirken, würden alle Objekte zum Massemittelpunkt gezugen werden und die Galaxie würde somit implodieren. Es ist also nötig auf die Sterne in der Galaxie anfangskräfte auszuwirken. Diese Kräfte sind durch die Rotation der Galaxie um den Massemittelpunkt der Galaxie definiert, man rotiert also die Galaxie und gleicht dadurch die Kraft die Alle Sterne richtung Massemittelpunkt zieht aus. Rotiert man die Galaxie jedoch zu schnell, explodiert sie Förmlich, da die Sterne nicht mehr zusammengehalten werden und die Fliehkraft sie einfach auseinanderzieht.

4.2 Berechnung der Beschleunigungen

Um die Beschleunigung die auf einen Stern wirk zu berechnen wird folgendes berechnet:

$$a = G \cdot \frac{\Delta M_2}{\Delta r^2} \tag{4}$$

G steht hier fúr die Universellt Gravitaionskraft, ΔM für die Masse des Objektes das Umkresit wird und Δr für die Entfernung zum Mittelpunkt des Objektes das umkreist wird. Problem ist, dass kein Objekt umkreist wird sondern eine große Anzahl an Sternen. Es ist also nich möglich mithilfe von herkömmlichen Methoden die Beschleunigung die auf einen Stern wirkt zu berechnen.

[2] https://de.wikipedia.org/wiki/Galaxie

4.2.1 Die Kraft als Vektor

Um die Kraft als Vektor darkzustellen, muss die Formel 4 mithilfe von Vektoren neu aufgestellt werden:

$$\vec{F}_{12} = \underbrace{-G\frac{m_1 m_2}{|r_{12}|^2}}_{Scalar} \cdot \underbrace{\frac{r_2 - r_1}{|r_2 - r_1|}}_{Vector} \tag{5}$$

Die Summe der Kräfte die auf einen Stern wirken ist somit die Summe aller Kräfte die zwischen dem jeweiligen Stern a und allen anderen Sternen wirken:

$$F_a = \sum_{i=0}^{n-1} F_{ai} \tag{6}$$

4.2.2 Probleme

Ein Problem das auftritt wenn die Kräfte zwischen allen Sternen berechnet werden ist, dass der Rechenaufwand $O(n \cdot n - 1) \approx O(n^2)$ beträgt. Problematisch wird es, wenn der mittlere Fehler, der bei der Berechnung der Kraft entsteht, größer als die Kraft wird. Das passier bei Sternen die sehr weit von einander entfernt liegen. Statt weiterzurechnen kann man die Sterne dann einfach weglassen, da die Daten unzuverlässig sin.

Die Lösung des Problems ist die verwendung des Barnes-Hut Algorithmuses, der duch die Unterteilung der Galaxie in verschiden große Zellen die Rechenkomplexität von $O(n^2)$ auf $O(n \log(n)$ runterbricht:

4.2.3 Berechnung der Umlaufgeschwindigkeit

Die Umlaufgeschwindigket kann berechnet werden, indem die Kraft die die Sterne in die Mitte der Galaxie zieht $\left(F = G \cdot \frac{m \cdot M}{r^2}\right)$ mit der Zentripetalkraft $\left(F_z = \frac{m \cdot v^2}{r}\right)$ gleichgesetzt wird:

$$v = \sqrt{\frac{G \cdot M_E}{r}} \tag{7}$$

M_E steht dabei für die Masse der Erde, G für die Gravitaitonskonstante und r für den Bahnradius. Da wir jedoch nicht in der Erdumlaufbahn, sondern in einer Galaxienumlaufbahn hantieren, können wir nicht die Masse der Erde nutzen. Wir müssen daher eine andere Möglichkeit nutzen, die größe der Masse, die den Stern in Richtung Massemittelpunkt zieht zu berechnen.

4.2.4 Ellipsen und die Geschwindigkeit der Sterne

Da die Sterne nicht auf perfekten Kreisbahnen um den Mittelpunkt der Galaxie fliegen muss in betracht gezogen werden wie die Sterne auf Elliptischen Bahnen orbitieren. Wichtigt ist dabei die Geschwindigkeit, diese muss zwischen der ersten Kosmischen Geschwindigkeit v_k und der zweiten Kosmischen Geschwindigkeit v_P liegen. Die beiden Kosmischen Geschwindigkeiten sind folgendermaßen definiert:

$$v_{k1} = \sqrt{\frac{GM}{r}} \tag{8}$$

$$v_{k2} = \sqrt{\frac{2GM}{r}} \tag{9}$$

Die Tatsache das die Sterne auf Elliptischen Bahnen unterwegs sind ist für die Berechnung irrelevant, da eh für jeden Zeitschritt t eine neue Kraft berechnet wird aus der eine Beschleunigung berechnet wird die wiederum die neue Position des Sternes ergibt. Hält man die Geschwindigkeit der Sterne somit im interval $(v_{k1}; v_{k2})$, dann ergibt sich (in der Therorie) von alleine eine elliptische Bahn.

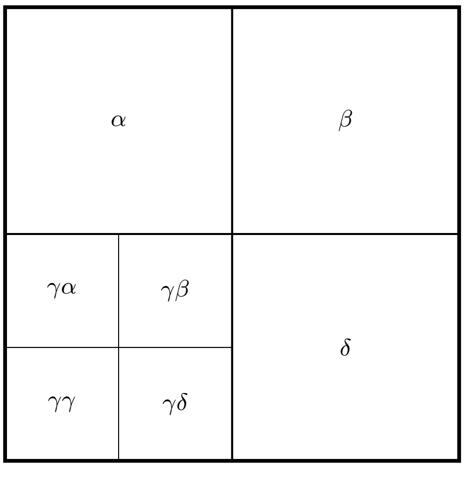
4.3 Entwicklung der nötigen Software

Die Software ist komplett in Golang geschrieben was die nutzung von mehreren Threads mithilfe von Go-Methoden stark vereinfacht. Um den Barnes-Hut Algorithmus anzuwenden muss die Galaxie in einen Octree unterteilt werden. Dabei wird eine Zelle definiert die alle Sterne beinhaltet welche anschließen solange unterteilt, bis eine der drei Endbedingungen eintrifft:

- Die Zelle enthält weniger als eine vordefinierte mindestmenge an Sternen
- Die Zelle ist kleiner als eine vordefinierte mindestgröße
- Es wurde eine maximale Anzahl an Unterteilungen vorgenommen

Ein wichtiger Aspekt ist jedoch auch, dass die Zellen rekursiv generiert werden. Kurzgesagt, die 'kinder'-Zellen dürfen müssen aus der Koordinates der 'Eltern'-Zellen generiert werden. Ist eine die übergeordnete Zell beispielweise definiert durch ihren Mittelpunkt m und die Maximale Breite b des Feldes, kann die Position der jeweiligen untergeordneten Zelle folgendermaßen berechnet werden:

$$NW = \left(m \pm \frac{b}{2}, m \pm \frac{b}{2}\right) \tag{10}$$



Nehmen wir als Beispiel das Feld $\gamma\beta$.

4.3.1 Barnes-Hut-simulation

Wie bereits beschrieben wird die Galaxie in Zellen unterteilt. Dabei kann, wenn eine Zelle weit genug von einem spezifischen Stern entfernt ist, die Zelle zu ihrem Massemittelpunkt vereinfacht werden. Der Massemittelpunkt kann wiefolgt berechnet werden:

$$\left(\frac{\sum_{i=0}^{n} x_i \cdot m_i}{\sum_{i=0}^{n} m}, \frac{\sum_{i=0}^{n} y_i \cdot m_i}{\sum_{i=0}^{n} m}\right) \tag{11}$$

Dabei wird die mithilfe ihrer Masse gewichtete Summe der Sterne durch die gesamt Masse geteilt um die geweilige Coordinaten-komponente zu erhalten. Dies muss für jede Zelle einmal getan werden und in der jeweiligen Zellen-struktur gespeichert werden wodurch am Ende jede Zelle die Koordinaten ihres Massemittelpunktes kennt.

Der Barnes-Hut-Algorithmus verringert die Anzahl zu berechnenden Kräfte durch geeignetes Zusammenfassen von Teilchengruppen zu Pseudoteilchen [3].

Der um eine Abschätzung zu bekommen, wie gut es ist, die Sterne zu bündeln, muss darauf geachtet werden, das das Verhältnis vom Gruppendurchmesser d zum Radius r möglichst gering ist:

$$\theta = \frac{d}{r} \tag{12}$$

Die Datenstruktur die einen Barnes-Hut Baum speichert ist am besten wiefolgt definiert:

4.3.2 Datentypen

Um generell irgendwas zu tun muss in fast allen fällen etwas definiert werden. Zur Simulation von Galaxien brauchen wir voallem eine Methode, Sterne einheitlich zu definieren. Der unten definierte Vec2-typ kann einen Vektor oder eine Koordinate darstellen was ihn in der Anwendung zu einem praktischen Hilfsmittel macht. Er speichert die X und Y Komponente der jeweiligen Struktur die er darstellen soll als float64-typ.

```
type Vec2 struct {
        X                float64
        Y                float64
}
```

Mithilfe des Vec2-typens kann ein Kompletter Stern definiert werden. Dabei wird ein Stern mithilfe seine Position C, seiner Geschwindigkeit V, und seiner Masse M beschrieben.

```
type Star2D struct {
        C                Vec2
        V                Vec2
        Mass             float64
}
```

Um einen sogenannten quadtree bzw. octree aufzubauen wird erstmal eine Räumliche Begrenzung benötigt, die einem Raum beschriebt indem die Sterne enthalten sind oder nicht. Diese grenze ist als ‘Boundary‘ definiert, es wird dabei der Mittelpunkt der Begrenzung und die kürzeste Entfernung zwischen mittelpunkt und äußerer Begrenzung genutzt um den Raum zu definieren.

```
type Boundary struct {
        Center           Vec2
        HalfDimension    float64
}
```

Der eigentliche QuadTree bzw. Octree beinhaltet einige Informationen: Die Anzahl in ihm enthaltene Sterne, die Räumliche ausbreitung, die eigentlichen Sterne als Star2D definiert und die RecursionTiefe als integer. Die Definition des QuadTrees der Unten zu sehen ist enthält Zeiger zu den Kindern des Quadtrees und ist somit rekusriv definiert was es einfach macht neue Kinder zu erstellen, da diese eine Kopie ihere Eltern mit einer anderen Begrenzung darstellen wodurch die in ihnen enthaltenen Sterne weniger werden.

```
type QuadTree struct {
        StarCount        int
        Boundary         Boundary
        Star             []Vec2

        NorthWest        *QuadTree
        NorthEast        *QuadTree
        SouthWest        *QuadTree
        SouthEast        *QuadTree

        ReccursionDepth  int
}
```

Idee: Wenn man bei herrausfinden welcher Stern in welcher Zelle liegt jedem Stern eine Zellen-id zuweist, kann man wenn man die Kraft zwischen zwei sehr weit entfernten Sternen berechnen will direkt dazu übergehen, die Kraft zum Massemittelpunkt der Zelle indem der weit eintferne Stern liegt zu berechnen.

4.3.3 Runge-Kutta methods

Die Runge-Kutta Methode wird genutzt, um die Position eines objektes nach einer Beliebigen Zeit zu approximieren. Dabei kann, bei nutzung eines mglich kleinen Zeitschrittes, ein sehr genaues Ergebniss erzielt werden. In unserem Fall haben wir einen Stern auf den eine Kraft wirkt. Wir wollen die Position des Sternens nach einem Zeitschritt berechnen, jedoch auch eine andere Kraft miteinbringen um die Sterne auf eine Ellipische Bahn um die Mitte der Galaxie zu bringen. Die Geschwindigkeit die der Stern dabei annnimmt kann mit der fogenden Formel berechnet werden:

$$v = \sqrt{ar} \tag{13}$$

4.3.4 Goroutines

Die nutzung von mehreren sogennanten Go-Methoden ist unglaublich effektiv, da es die Zeit die gebraucht wird das Programm auszuführen drastisch verringert. Die implementation ist ebenfalls unglaublich einfach, es recht

5 Ergebnisse

Wie bewertest Du Deine Ergebnisse? Wie passt das zu dem, was Du über Dein Thema gelesen oder gehört hast? Was ist gut gelaufen im Projekt, was war schlecht, was könnte noch verbessert werden? Das simulieren von Galaxien ist komplett ohne optimierungen ein sehr rechenaufwendiger prozess, weshalb einer der wichtigsten aspekte des Projektes war, die effizienz zu erhöhen.

5.1 Das n-Körper Problem

Das N-Körper Problem ist blöd (aber notwendig D:) (Und es ermöglicht das ganze erst!)

[3] https://de.wikipedia.org/wiki/Barnes-Hut-Algorithmus

5.2 Beschleunigung der Berechnung von Kräften

$n^2 \to n \cdot log(n)$ iasd

5.3 Fazit

Welche Antwort kannst Du auf Deine Forschungsfrage geben? Hast Du Dein Ziel erreicht? Langsam, Ok, schneller, Schnell, Hyperspeed!

6 Quellen und Literaturverzeichnis

THE INTERNET!