

Zusammenfassung

Die vorliegende Bachelorarbeit gliedert sich in die beiden Teile *Theoretische Grundlagen* und *Praktische Ausführung*. In ersterem werden die Grundlagen des Rejection Sampling und des Markov-Chain-Monte-Carlo Verfahrens erläutert, die anschließend im zweiten Teil zur Lösung des vorliegenden Problems anwendung finden, welches in der Stabilisierung eines physikalischen Körpers im Raum besteht. Um die Effizienz der Algorithmen angemessen bewerten zu können, werden mehrere Versuche mit variierenden Parametern durchgeführt. Die Versuchsergebnisse werden anschließend ausgewertet, analysiert und visualisiert.

Inhaltsverzeichnis

I. Theoretischer Hintergrund	5
1. Rejection Sampling	6
1.1. Grundlagen[1]	6
1.2. Algorithmus[1]	7
2. Markov-Chain-Monte-Carlo-Verfahren	8
2.1. Grundlagen	8
2.1.1. Markov-Ketten[1]	8
2.1.2. MCMC Grundlagen[1]	10
2.1.3. Metropolis-Hastings Sampling[1]	11
2.2. Algorithmus[1]	13
II. Praktische Ausführung	15
3. Ausgangssituation	16
3.1. Problem 1	16
3.2. Problem 2	16
3.3. Formulierung des Problems	17
3.4. Zu berücksichtigende Gegebenheiten der Simulationsumgebung	18
4. Rejection Sampling	21
4.1. Problemstellung im Rejection Sampling	21
4.2. Algorithmus	22
4.3. Anwendung auf Problem 1	22
4.3.1. Versuchsparameter	22
4.3.2. Ergebnisanalyse	23
4.3.3. Visualisierung der Versuchsergebnisse	24
4.4. Anwendung auf Problem 2	25
4.4.1. Versuchsparameter	25
4.4.2. Ergebnisanalyse	26
4.4.3. Visualisierung der Versuchsergebnisse	26

5. MCMC-Verfahren	34
5.1. Problemstellung in MCMC	34
5.2. Algorithmus	34
5.3. Anwendung auf Problem 1	35
5.3.1. Versuchsparameter	35
5.3.2. Ergebnisanalyse	36
5.3.3. Visualisierung der Versuchsergebnisse	37
5.4. Anwendung auf Problem 2	37
5.4.1. Versuchsparameter	38
5.4.2. Ergebnisanalyse	38
5.4.3. Visualisierung der Versuchsergebnisse	39
6. Optimiertes MCMC	45
6.1. Optimierung für Problem 1	45
6.1.1. Versuchsparameter	46
6.1.2. Ergebnisanalyse	46
6.1.3. Visualisierung der Versuchsergebnisse	47
6.2. Optimierung für Problem 2	47
6.2.1. Versuchsparameter	47
6.2.2. Ergebnisanalyse	47
6.2.3. Visualisierung der Versuchsergebnisse	49
6.2.4. Zweite Versuchsreihe	49
6.2.5. Ergebnisse	49
7. Vergleich aller Ergebnisse	54

Einleitung

Inhalt der vorliegenden Arbeit ist die Erarbeitung von Algorithmen, mit deren Hilfe sich einfache physikalische Problemstellung über das Ziehen von Proben aus Wahrscheinlichkeitsverteilungen lösen lassen.

Dabei nähern wir uns dem Problem zunächst mittels Rejection Sampling an, um anschließend die Markov-Chain-Monte-Carlo Methode darauf anzuwenden, bevor zuletzt die Ergebnisse beider Verfahren bezüglich diverser Kriterien miteinander verglichen werden.

Dabei interessieren uns insbesondere zwei bestimmte Problemstellungen:

- Problem 1: Das Stabilisieren eines Körpers im Raum in einer vorgegebenen Höhe über dem Boden
- Problem 2: Das Stabilisieren eines Körpers im Raum in einer vorgegebenen Höhe über einem Fluss mit zwei Ufern.

Der erste Teil *Theoretischer Hintergrund* der vorliegenden Arbeit enthält die den Versuchen zu Grunde liegenden theoretischen Grundlagen. In Teil zwei, *Praktische Ausführung*, werden auf diesen Grundlagen basierende Algorithmen zur Lösung der Problemstellungen entworfen.

Diese Algorithmen werden anschließend in einer simulierten physikalischen Umgebung getestet. Die Ergebnisse werden einer Analyse unterzogen und entsprechend visualisiert, bevor schließlich eine vergleichende Analyse der Ergebnisse den Abschluss bildet.

Teil I.
Theoretischer Hintergrund

1. Rejection Sampling

1.1. Grundlagen[1]

Rejection Sampling bietet eine simple Methode, um Proben aus einer Wahrscheinlichkeitsverteilung

$$p(x) = \frac{1}{Z} p^*(x) \quad (1.1)$$

zu ziehen, die bis auf die Konstante Z bekannt ist und für die keine bekannte Methode für das Ziehen der Proben existiert. Voraussetzung hierfür ist, dass man eine andere Verteilung $q(x)$ kennt, aus der sich - nach Möglichkeit effizient - Proben ziehen lassen.

Wir werden nun $q(x)$ verwenden, um Proben aus $p(x)$ zu ziehen. Zu diesem Zweck führen wir die binäre Variable $y \in \{0, 1\}$ ein und definieren des Weiteren die Verteilung

$$q(x, y) := q(y|x)q(x), \quad (1.2)$$

wobei wir festlegen, dass

$$q(y = 1|x) \propto \frac{p(x)}{q(x)}. \quad (1.3)$$

gilt. Da $p(x)$ und $p^*(x)$ sich lediglich durch die Konstante Z unterscheiden, folgt hieraus direkt $q(y = 1|x) \propto \frac{p^*(x)}{q(x)}$. Da uns $p^*(x)$ und $q(x)$ bekannt sind, lässt sich diese bedingte Verteilung daher ohne Weiteres generieren.

Multipliziert man Gleichung 1.3 mit $q(x)$ und setzt 1.2 ein, so ergibt sich

$$q(x, y = 1) \propto p(x). \quad (1.4)$$

Damit ist gezeigt, dass wir eine von x abhängige Verteilung $q(x, y)$ generieren können, die proportional zu unserer gewünschten Verteilung $p(x)$ ist und aus der wir über $q(x)$ Proben ziehen können.

Im nächsten Schritt bestimmen wir $q(y|x)$ genauer und führen eine Konstante M ein, die verhindert, dass wir Übergangswahrscheinlichkeiten größer als 1 erhalten. Zu diesem Zweck wählen wir M so, dass

$$q(y|x) = \frac{p^*(x)}{M \cdot q(x)} \leq 1 \quad (1.5)$$

für alle x gilt. Somit ist $q(y|x)$ normalisiert.

1.2. Algorithmus[1]

Im Folgenden entwerfen wir einen Algorithmus, in dem wir die oben definierten Grundlagen zur Anwendung bringen und mit dem wir Proben aus unserer Verteilung $q(x, y) \propto p(x)$ ziehen können.

Zunächst bestimmen wir ein M , so dass die Bedingung 1.5 erfüllt ist. An dieser Stelle wollen wir vorerst annehmen, dass wir eine effektive Methode kennen, mit der sich ein solches M bestimmen lässt. In Kapitel 4.1 wird eine Methode eingeführt, mit der sich ein M für die vorliegende Problemstellung bestimmen lässt. Sollte sich während der Ausführung des Algorithmus zeigen, dass M die Bedingung wider Erwarten nicht für alle x erfüllt, so wird der Wert korrigiert und der Algorithmus beginnt von Vorne.

Nehmen wir nun an, dass wir L Proben aus $p(x)$ ziehen wollen. Dabei wählen wir L je nach Fall und Problemstellung so groß, dass wir eine hinreichend genaue Abbildung von $p(x)$ erhalten. Sollte sich am Ende des Versuches herausstellen, dass L zu niedrig gewählt war, korrigiert man den Wert entsprechend nach oben und lässt den Versuch weiterlaufen.

Um aus $q(x, y) = q(y|x)q(x)$ Proben zu ziehen wird zunächst eine Probe x^{probe} aus $q(x)$ gewählt. Diese Probe wiederum wird gemäß 1.2 mit der Wahrscheinlichkeit $q(y = 1|x)$ akzeptiert. Andernfalls wird die Probe verworfen und ein neues x^{Probe} aus $q(x)$ gezogen.

Zur Umsetzung dieses Auswahlverfahrens wird ein auf einem Zufallsgenerator basierendes Verfahren benötigt. Der Wert $a = q(y = 1|x)$ lässt sich direkt über 1.5 bestimmen. Anschließend wird gleichverteilt ein zufälliges u aus dem Intervall $[0, 1]$ gezogen. Gilt $u \leq a$, so wird x^{probe} als Probe aus $p(x)$ akzeptiert- andernfalls wird es verworfen.

Die Repräsentation dieses Algorithmus als Pseudocode findet sich in Algorithm 1 auf Seite 7.

Algorithm 1 Verwerfungsmethode

- 1: Zu gegebenem $p^*(x)$ und $q(x)$ ein M finden, so dass $\frac{p^*(x)}{q(x)} \leq M$ für alle x erfüllt ist
 - 2: **for** $l = 1$ bis L **do**
 - 3: **repeat**
 - 4: Probe x^{probe} aus $q(x)$ ziehen
 - 5: Sei $a = \frac{p^*(x)}{M \cdot q(x)}$
 - 6: Ziehe gleichverteilt ein $u \in [0, 1]$
 - 7: **until** $u \leq a$
 - 8: $x^l = x^{probe}$
-

2. Markov-Chain-Monte-Carlo-Verfahren

2.1. Grundlagen

2.1.1. Markov-Ketten[1]

Zum Verständnis des Markov-Chain-Monte-Carlo-Verfahren, im Weiteren als MCMC-Verfahren abgekürzt, bietet es sich an, an dieser Stelle Markov-Übergänge und Markov-Ketten einzuführen.

Sei x eine Zufallsvariable, die zu diskreten Zeitpunkten t einen der Zustände x_1, x_2, \dots, x_n annimmt, wobei wir die jeweilige Zeit, zu der ein bestimmter Zustand angenommen wird, mit einem hochgestellten Index bezeichnen. So ist zum Beispiel x_5^3 der fünfte der gegebenen Zustände, der zum Zeitpunkt $t = 3$ eingenommen wird.

Die Wahrscheinlichkeit, in einem Zeitschritt von Zustand x_i zu x_j überzugehen wird mit p_{ij} bezeichnet. Beispielsweise wird die Wahrscheinlichkeit, von x_5 zum Zeitpunkt t_1 zu x_2 zum nachfolgenden Zeitpunkt $t_2 = t_1 + 1$ überzugehen, mit p_{52} bezeichnet.

Für unsere Zwecke ist es ausreichend, zeitinvariante Übergangswahrscheinlichkeiten zu betrachten. Die verschiedenen p_{ij} sind in diesem Fall also für alle Zeiten gleich. Markov-Ketten, deren Übergangswahrscheinlichkeiten diese Eigenschaft besitzen, werden als stationär bezeichnet.

In Abbildung 2.1 ist eine grafische Veranschaulichung eines Markov-Modells mit drei Zuständen x_1, x_2, x_3 und den jeweiligen Übergangswahrscheinlichkeiten zu sehen.

Wir untersuchen nun, wie hoch die Wahrscheinlichkeit ist, zu einem beliebigen Zeitschritt t^* bei einem bestimmten Zustand x_j zu sein. Es ist offensichtlich, dass die vorhergehenden Zustände X^1, \dots, X^{t^*-1} diese Wahrscheinlichkeit beeinflussen. Es gilt also $p(x_j^{t^*}) = p(x_j^{t^*} | X^{t^*-1}, \dots, X^1)$, wobei wir aus der Theorie der Markov-Ketten übernehmen, dass Zustände in naher Vergangenheit einen größeren Einfluss auf unseren aktuellen Zustand haben, als solche, die weiter zurückliegen. Daher bietet es sich an, nur eine bestimmte Anzahl vergangener Zustände bei der Bestimmung der Wahrscheinlichkeit zu berücksichtigen. Wir sagen, eine Markov-Kette ist vom Grad n , wenn die Zustände der letzten n Zeitpunkte bei der Berechnung der Wahrscheinlichkeit berücksichtigt werden. In einer Markov-Kette ersten Grades wird so beispielsweise nur der Zustand des unmittelbar vorhergehenden Zeitpunktes $t^* - 1$ berücksichtigt und es gilt

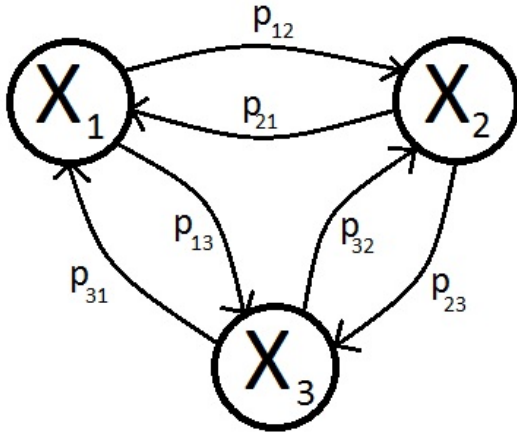


Abbildung 2.1.: Markov Modell

$$p(X^{t^*} | X^{t^*-1}, \dots, X^1) = p(X^{t^*} | X^{t^*-1}). \quad (2.1)$$

Wir veranschaulichen die Bestimmung der stationären Wahrscheinlichkeit, zu einem bestimmten Zeitpunkt bei Zustand x_j zu sein, an einer diskreten Markov-Kette ersten Grades. Die allgemeine Wahrscheinlichkeit, zum Zeitpunkt t^* bei x_j zu sein, ergibt sich zu

$$p(X^{t^*} = x_j) = \sum_{x_i} p(X^{t^*} = x_j | X^{t^*-1} = x_i) p(X^{t^*-1} = x_i). \quad (2.2)$$

In Worten ausgedrückt bedeutet dies, wir multiplizieren die Wahrscheinlichkeit, zum Zeitpunkt $t - 1$ bei x_i zu sein, mit der Wahrscheinlichkeit $p_{ij} = p(X^{t^*} = x_j | X^{t^*-1} = x_i)$, von x_i zu x_j überzugehen und summieren über alle Zustände x_i . Diese Wahrscheinlichkeit ist Schwankungen unterworfen, da die Wahrscheinlichkeiten $p(x_i)$ voneinander abhängig sind und sich somit gegenseitig beeinflussen.

Um dies zu veranschaulichen, betrachten wir unser in Abbildung 2.1 dargestelltes Modell. Wir legen fest, dass wir bei x_1 beginnen, dass also gilt $X^1 = x_1$. Daraus folgt unmittelbar $p(X^1 = x_1) = 1$, $p(X^1 = x_2) = 0$ und $p(X^1 = x_3) = 0$.

Nun legen wir fest, dass $p_{12} = 0,4$ und $p_{13} = 0,6$ gilt. Damit ergibt sich $p(X^2 = x_1) = 0$, $p(X^2 = x_2) = 0,4$ und $p(X^2 = x_3) = 0,6$. Man erkennt an diesem Beispiel, dass die Wahrscheinlichkeit, zu einem bestimmten Zeitpunkt bei einem bestimmten Zustand zu sein, stark zeitabhängig sein kann.

Beobachtet man eine Markov-Kette jedoch über einen längeren Zeitraum, so bildet sich ein Gleichgewicht mit stationären Wahrscheinlichkeiten für die einzelnen Zustände heraus. Diese stationären Wahrscheinlichkeiten werden im Folgenden bestimmt.

Zu diesem Zweck führen wir den Vektor

$$\mathbf{v}(t) = \begin{pmatrix} p(x_1, t) \\ p(x_2, t) \\ \vdots \\ p(x_n, t) \end{pmatrix}, \quad (2.3)$$

der die einzelnen Wahrscheinlichkeiten der Zustände enthält und zeitabhängig ist, sowie die Matrix

$$\mathbf{M} = \begin{pmatrix} p_{11} & p_{21} & \cdots & \\ p_{12} & p_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ & & & p_{nn} \end{pmatrix} \quad (2.4)$$

der konstanten Übergangswahrscheinlichkeit, ein. So können wir die Verteilung der Wahrscheinlichkeiten zum Zeitpunkt t^* über eine einfache Matrixoperation berechnen:

$$\mathbf{v}(t^*) = \mathbf{M} \cdot \mathbf{v}(t^* - 1). \quad (2.5)$$

Die stationäre Verteilung der Wahrscheinlichkeiten ist erreicht, wenn sich die einzelnen Komponenten von \mathbf{v} nicht mehr ändern, wenn also $\mathbf{v}(t^*) = \mathbf{v}(t^* - 1)$ gilt. Diese Eigenschaft weisen die Eigenvektoren $\mathbf{v}_E(t)$ von \mathbf{M} auf, für die gilt

$$\mathbf{M} \cdot \mathbf{v}_E(t) = a \cdot \mathbf{v}_E(t), \quad (2.6)$$

wobei a der zum jeweiligen Eigenvektor gehörende Eigenwert ist. Da $\mathbf{v}(t)$ und somit $\mathbf{v}_E(t)$ per Definition stets normiert ist (Die Gesamtwahrscheinlichkeit darf nicht größer als 1 werden), folgt unmittelbar $a = 1$.

Der zu bestimmende Vektor \mathbf{v}_E enthält somit die gesuchte stationäre Wahrscheinlichkeitsverteilung der Zustände x_1, \dots, x_n , wobei $\mathbf{v}_E(t)$ jedoch nicht eindeutig sein muss.

2.1.2. MCMC Grundlagen[1]

Das MCMC-Verfahren eignet sich, um Proben aus einer bis auf die Normierungskonstante Z bekannte Verteilung $p(x) = \frac{1}{Z}p^*(x)$ zu ziehen. Dies ist gerade in denjenigen Fällen von Interesse, in denen sich Z nicht bestimmen lässt.

Zu diesem Zweck benötigen wir eine uns bekannte Verteilung $q(x)$, aus der wir möglichst effizient Proben ziehen können und die dieselbe stationäre Verteilung wie $p(x)$

aufweist. Die Forderung der identischen stationären Verteilung liefert die Garantie, dass das Ziehen aus $q(x)$ äquivalent zum Ziehen aus $p(x)$ ist, wenn die Anzahl der gezogenen Proben gegen unendlich geht.

Da $q(x)$ im Allgemeinen nicht die selbe stationäre Verteilung wie $p(x)$ hat, wollen wir uns nun mittels eines Markov-Übergangs eine Verteilung konstruieren, die diese Eigenschaft aufweist. Die hier behandelte Methode ist unter dem Namen Metropolis-Hastings Sampling bekannt.

2.1.3. Metropolis-Hastings Sampling[1]

An dieser Stelle führen wir einen auf $q(x)$ basierenden Markov-Übergang ein, der wie gefordert die selbe stationäre Verteilung wie $p(x)$ besitzt. Zu diesem Zweck steht uns aus dem für das MCMC-Verfahren geeignete Metropolis-Hastings Verfahren der Übergang

$$q(x'|x) = \tilde{q}(x'|x)f(x', x) + \delta(x', x)\left(1 - \int_{x''} \tilde{q}(x''|x)f(x'', x)\right) \quad (2.7)$$

zur Verfügung.

Im Folgenden bestimmen wir die einzelnen Komponenten $\tilde{q}(x'|x)$ und $f(x', x)$ näher. Bei $\delta(x', x)$ handelt es sich um das Kronecker-Delta mit der Eigenschaft

$$\delta(x', x) = \begin{cases} 1, & \text{falls } x' = x \\ 0, & \text{sonst} \end{cases} \quad (2.8)$$

Bevor wir mit der näheren Bestimmung von $\tilde{q}(x'|x)$ und $f(x', x)$ fortfahren, zeigen wir noch, dass es sich bei $q(x'|x)$ um eine gültige Verteilung handelt, wenn wir voraussetzen, dass es sich bei $\tilde{q}(x'|x)$ um eine gültige Verteilung handelt und dass stets $f(x', x) \in [0, 1]$ gilt.

Satz. $q(x'|x)$ ist eine gültige Verteilung.

Beweis. Teil 1: Nichtnegativität von $q(x'|x)$

Die Nichtnegativität von $q(x'|x)$ folgt für $x' \neq x$ direkt aus der Definition unserer Funktion 2.7, da sowohl $f(x', x)$, als auch $\tilde{q}(x'|x)$ per Definition stets nichtnegativ sind. Für den Fall $x' = x$, in dem unsere aktuell gezogene Probe mit der vorhergehenden identisch ist, erhalten wir durch Einsetzen

$$q(x|x) = \tilde{q}(x|x)f(x, x) + \delta(x, x)(1 - \int_{x''} \tilde{q}(x'', x)f(x'', x)) \quad (2.9)$$

$$= 1 + \tilde{q}(x|x)f(x, x) - \int_{x''} \tilde{q}(x'', x)f(x'', x) \quad (2.10)$$

$$\geq 1 + \tilde{q}(x|x)f(x, x) - 1 \quad (2.11)$$

$$= \tilde{q}(x|x)f(x, x) \geq 0. \quad (2.12)$$

Teil 2: Es gilt $\int_{x'} q(x'|x) = 1 \leftrightarrow$ die Fläche unter $q(x'|x)$ ist genau 1.

$$\int_{x'} q(x'|x) = \int_{x'} \left[\tilde{q}(x'|x)f(x', x) + \delta(x', x)(1 - \int_{x''} \tilde{q}(x'', x)f(x'', x)) \right] \quad (2.13)$$

$$= \int_{x'} \tilde{q}(x'|x)f(x, x') + 1 - \int_{x''} \tilde{q}(x'', x')f(x'', x') = 1 \quad (2.14)$$

Damit ist gezeigt, dass $q(x'|x)$ alle Kriterien einer Verteilung erfüllt und wir diese somit in diesem Sinne verwenden können. \square

Im Weiteren suchen wir eine Funktion $f(x', x)$, die garantiert, dass $q(x'|x)$ für jedes $\tilde{q}(x'|x)$ die selbe stationäre Verteilung wie $p(x)$ besitzt. Aus dieser Bedingung ergibt sich

$$p(x') = \int_x q(x'|x)p(x) \quad (2.15)$$

$$= \int_x \tilde{q}(x'|x)f(x', x)p(x) + p(x')(1 - \int_{x''} \tilde{q}(x''|x')f(x'', x)) \quad (2.16)$$

$$\Leftrightarrow \int_x \tilde{q}(x'|x)f(x', x)p(x) = \int_x \tilde{q}(x|x')f(x, x')p(x'), \quad (2.17)$$

wobei in der letzten Umformung $p(x')$ subtrahiert und x'' mit x substituiert wurde. Nun führen wir für $f(x', x)$ die Funktion

$$f(x', x) = \min \left(1, \frac{\tilde{q}(x|x')p(x')}{\tilde{q}(x'|x)p(x)} \right) = \min \left(1, \frac{\tilde{q}(x|x')p^*(x')}{\tilde{q}(x'|x)p^*(x)} \right) \quad (2.18)$$

ein, von der wir zeigen werden, dass sie unser gefordertes Kriterium erfüllt. Zur Bestätigung genügt es, $f(x', x)$ in 2.17 einzusetzen und ein wenig umzuformen:

$$\tilde{q}(x'|x)f(x', x)p(x) = \tilde{q}(x'|x) \cdot \min\left(1, \frac{\tilde{q}(x|x')p(x')}{\tilde{q}(x'|x)p(x)}\right)p(x) \quad (2.19)$$

$$= \min(\tilde{q}(x'|x)p(x), \tilde{q}(x|x')p(x')) = \min(\tilde{q}(x|x')p(x'), \tilde{q}(x'|x)p(x)) \quad (2.20)$$

$$= \tilde{q}(x|x') \cdot \min\left(1, \frac{\tilde{q}(x'|x)p(x)}{\tilde{q}(x|x')p(x')}\right)p(x') = \tilde{q}(x|x')f(x, x')p(x'). \quad (2.21)$$

Hieraus folgt unmittelbar die Gleichheit der beiden Integrale aus Gleichung 2.17 und daraus wiederum die Gleichheit der stationären Verteilungen von $p(x')$ und $q(x'|x)$.

Da diese aus dem Metropolis-Hastings-Verfahren stammende Verteilung $q(x'|x)$ wie gezeigt alle geforderten Kriterien erfüllt, wollen wir nun einen Algorithmus entwerfen, mittels dem wir Proben aus diesem Markov-Übergang ziehen wollen, um $p(x)$ zu approximieren. Je mehr Proben wir ziehen, desto näher kommen wir der stationären Verteilung von $q(x'|x)$ und somit auch derjenigen von $p(x')$.

2.2. Algorithmus[1]

Um einen effizienten Algorithmus für das Ziehen der Proben aus $q(x'|x)$ zu entwerfen, ist es hilfreich, unsere Verteilung als Linearkombination aus zwei Funktionen zu betrachten:

1. $\tilde{q}(x'|x)f(x', x)$
2. $\delta(x', x) \left(1 - \int_{x''} \tilde{q}(x''|x)f(x'', x)\right)$.

Dies kann man so interpretieren, dass eine Probe x^l aus $\tilde{q}(x^l|x^{l-1})$ mit der Wahrscheinlichkeit $f(x^l, x^{l-1})$ akzeptiert wird. Wird x^l nicht akzeptiert, so setzt man, dem zweiten Fall entsprechend $x^l = x^{l-1}$.

Das selbe Resultat ergibt sich, wie durch Umformen leicht zu erkennen ist, wenn wir x^l im Falle $\tilde{q}(x|x')p^*(x') \geq \tilde{q}(x'|x)p^*(x) \leftrightarrow \tilde{q}(x|x')p(x') \geq \tilde{q}(x'|x)p(x)$ akzeptieren, da sich aus der Erfüllung der Ungleichung nach 2.18 direkt $f(x', x) = 1$ ergibt. Ist die Ungleichung nicht erfüllt, so wird x^l mit einer Wahrscheinlichkeit von $\frac{\tilde{q}(x|x')p^*(x')}{\tilde{q}(x'|x)p(x)}$ als Probe akzeptiert. Wie auch schon im vorhergehenden Absatz gilt $x^l = x^{l-1}$ für den Fall, dass wir x^l als Probe nicht akzeptieren.

Die Umsetzung des Algorithmus in Pseudocode ist in Algorithm 2 auf Seite 14 einsehbar.

Algorithm 2 Metropolis-Hastings Sampling

```
1: Startprobe  $x^1$  wählen
2: for  $l = 2$  bis  $L$  do
3:   Probe  $x^l$  aus  $\tilde{q}(x^l|x^{l-1})$  ziehen
4:   Setze  $a = \frac{\tilde{q}(x^{l-1}|x^l)p(x^l)}{\tilde{q}(x^l|x^{l-1})p(x^{l-1})}$ 
5:   if  $a \geq 1$  then  $x^l$  wird als Probe akzeptiert
6:   else
7:     Ziehe gleichverteilt ein  $u$  aus der Intervall  $[0, 1]$ 
8:     if  $u \leq a$  then  $x^l$  wird als Probe akzeptiert
9:     else
10:      Setze  $x^l = x^{l-1}$ 
```

Teil II.
Praktische Ausführung

3. Ausgangssituation

An dieser Stelle übertragen wir die oben eingeführten theoretischen Grundlagen auf unsere konkreten Probleme, indem wir diese auf eine Art und Weise darstellen, die es uns erlaubt, die Prinzipien unserer eingeführten Sampling-Methoden anzuwenden.

Zuvorderst erläutern wir die eigentliche Problemstellung um sie anschließend soweit zu abstrahieren, dass wir die Algorithmen darauf anwenden und bezüglich diverser Kriterien [Güte der Lösung, Zeitaufwand, ...] testen und gegebenenfalls modifizieren können.

Die Umgebung, in der wir arbeiten, wird durch einen simulierten dreidimensionalen Raum dargestellt, in welchem wir, auf einer Gleichverteilung basierend, physikalische Objekte mit zufälliger Position, Orientierung und Form simulieren können, die dem Einfluss einer Gravitation unterworfen sind.

Nachfolgend werden die beiden grundlegenden Problemstellungen dargestellt, anhand derer wir arbeiten und die Sampling-Algorithmen testen.

3.1. Problem 1

Dieses Problem besteht darin, einen Körper in der Simulationsumgebung in einer vorgegebenen Höhe zu stabilisieren, also durch andere Körper abzustützen. Der Versuchsaufbau ist in Abbildung 3.1 (Seite 17) dargestellt. Als zu stabilisierenden Körper wählen wir einen Quader mit den Abmessungen $(1 \times 1 \times 0.25)$. Die Körper, die zur Stabilisierung dienen, sind ebenfalls Quader mit den Abmessungen $(1 \times 1 \times 0.25)$.

Unsere Aufgabe besteht darin, ein auf unseren Algorithmen basierendes Programm zu entwerfen, das eine Konstellation von Körpern findet, die den Quader stabilisiert. Dazu ist es nützlich, die Situation in einer Weise zu beschreiben, mit der wir arbeiten können. Dazu mehr in Kapitel 3.3.

3.2. Problem 2

Diese Problemstellung ist jener in *Problem 1* zwar ähnlich, bietet jedoch, wie wir im späteren Verlauf feststellen werden, einige Schwierigkeiten, die in Problem 1 nicht auftreten.

Kern des Problems ist, wie auch schon bei *Problem 1*, einen Körper im Raum zu stabilisieren. In diesem Fall legen wir jedoch fest, dass der Boden der Simulationsumgebung dabei nicht verwendet werden darf. Der hier zu stabilisierende Körper,

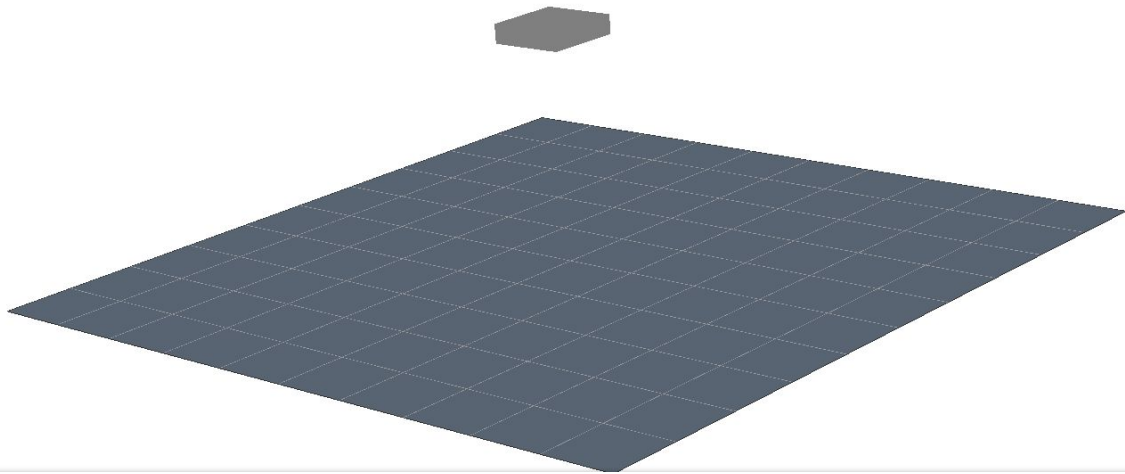


Abbildung 3.1.: Simulationsumgebung zu Problem 1

sowie die zur Stabilisierung simulierten Körper, sind Quader mit den Abmessungen $(1 \times 1 \times 0.1)$. An dieser Stelle bietet sich der Vergleich mit einem Fluss an:

Wenn wir einen Körper über einem Fluss stabilisieren wollen, so geschieht dies durch eine Brücke, die vom einen Ufer zum anderen reicht. Einen Turm, der den Körper stützt und im Wasser steht, wollen wir an dieser Stelle ausschließen. Der entsprechende Versuchsaufbau, in dem zwei Blöcke die Ufer darstellen, ist in Abbildung 3.3 zu sehen.

3.3. Formulierung des Problems

Im Hinblick auf die weiter oben angeführten Methoden wird der Zustand \mathbf{x} durch die zufällig simulierten Körper definiert, die eine stabile Konstellation ermöglichen sollen. Die einzelnen Zustände x_i beschreiben dabei die zur Beschreibung der Körper notwendigen Informationen. Dies sind hier drei Freiheitsgrade der Position, drei Freiheitsgrade der Rotation, sowie - im Falle von Quadern - drei Freiheitsgrade der Abmessungen.

Da Verteilungen über derart viele Freiheitsgrade nur schwer darstellbar sind und dies den Rahmen dieser Arbeit sprengen würde, reduzieren wir die Freiheitsgrade an dieser Stelle. Dazu wollen wir lediglich Quader mit festen Abmessungen betrachten, sowie nur Rotationen um die vertikale z -Achse zulassen. Somit verbleiben nur noch vier Freiheitsgrade: Drei in der Position und einer in der Rotation.

$p(\mathbf{x})$ soll bei uns angeben, wie hoch die Wahrscheinlichkeit ist, dass eine bestimmte Konstellation \mathbf{x} stabil ist, und wird in den folgenden Versuchen näher definiert.

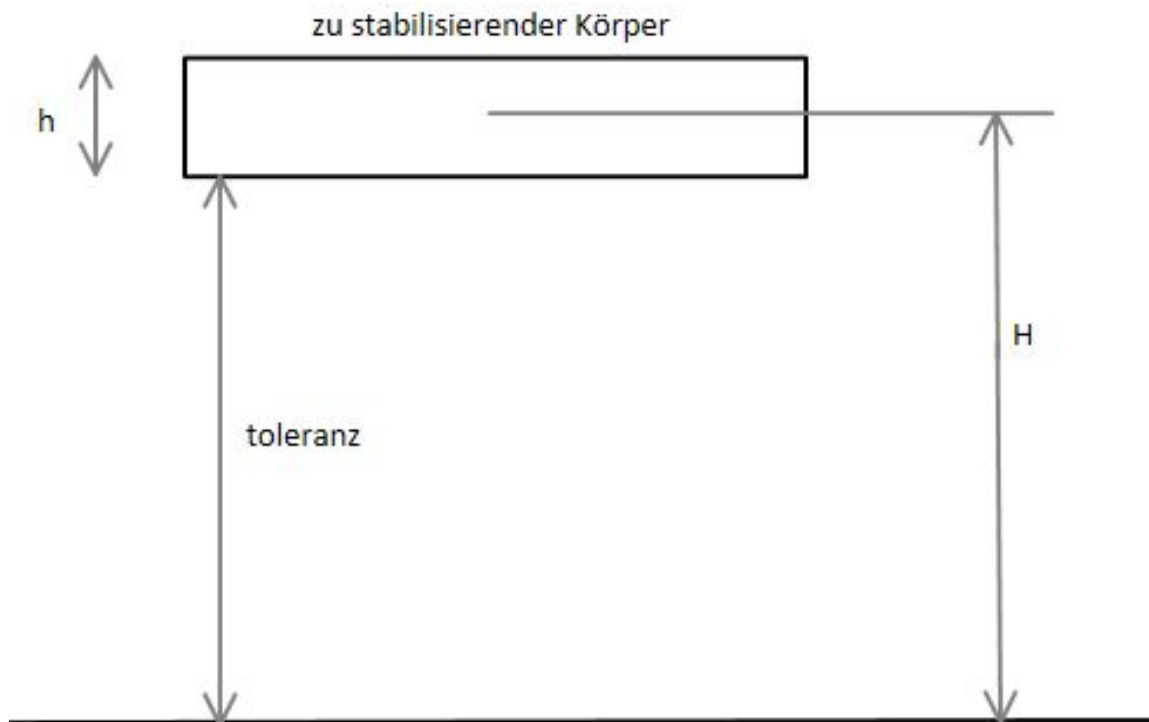


Abbildung 3.2.: Schematische Zeichnung zu Problem 1

3.4. Zu berücksichtigende Gegebenheiten der Simulationsumgebung

In der verwendeten Simulationsumgebung besteht die Möglichkeit, dass zufällig generierte Körper sich überschneiden. Dies führt dazu, dass diese sich gegenseitig abstoßen, was die Versuchsergebnisse in nicht unerheblichem Maße beeinflusst. Um diesen Einfluss zu minimieren, werden in allen Algorithmen Toleranzgrenzen eingeführt, die den Versuch abbrechen, sobald sich der zu stabilisierende Körper um eine größere Strecke fortbewegt, als dies im freien Fall zu erwarten wäre.

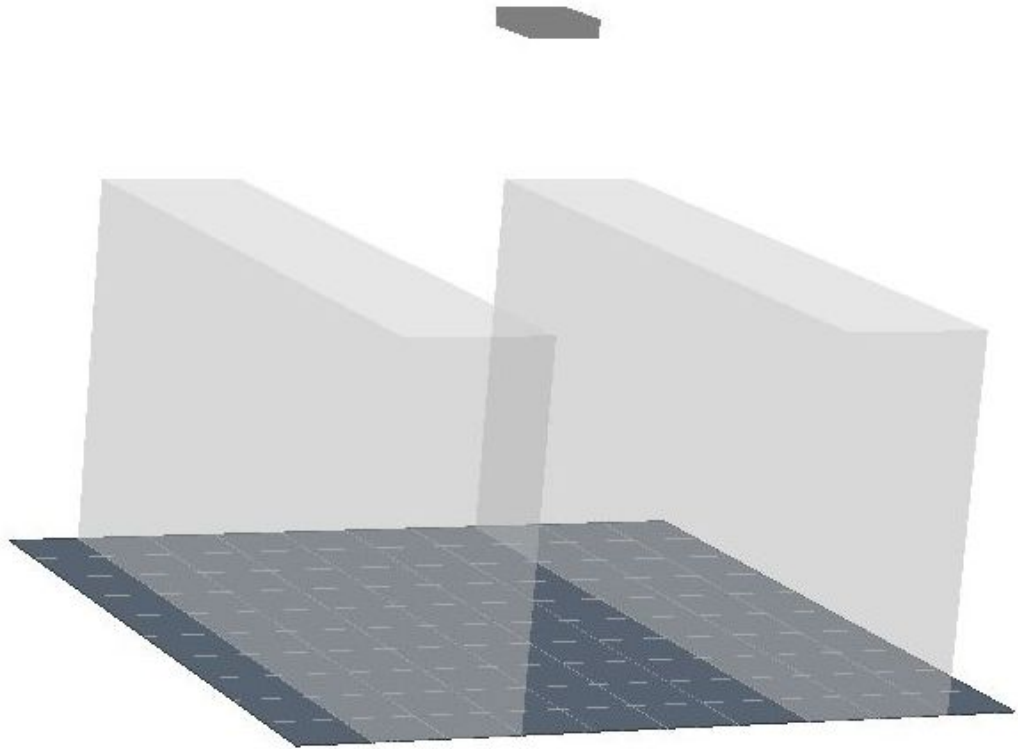


Abbildung 3.3.: Simulationsumgebung zu Problem 2

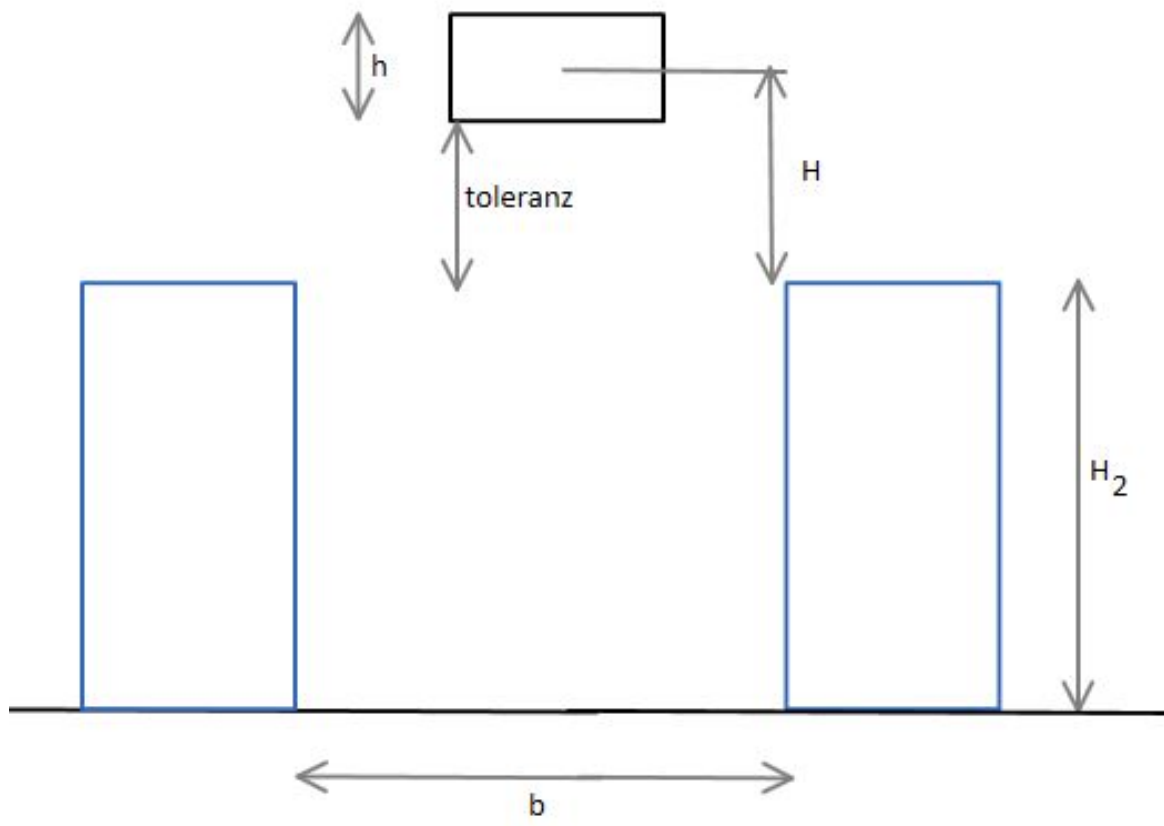


Abbildung 3.4.: Schematische Zeichnung zu Problem 2

4. Rejection Sampling

4.1. Problemstellung im Rejection Sampling

Wir approximieren die gesuchte Verteilung $p(\mathbf{x})$ mithilfe einer beliebigen anderen Verteilung $q(\mathbf{x})$. Die einfachste Variante bietet in diesem Fall die Gleichverteilung, so dass gilt $q(\mathbf{x}) = 1$, wobei wir annehmen, dass \mathbf{x} in der Art normiert ist, dass $\int q(\mathbf{x}) = 1$ gilt und wir somit eine gültige Verteilung erhalten.

Um $p(\mathbf{x})$ näher zu bestimmen, legen wir an dieser Stelle einige Variablen fest, die uns dabei helfen, eine auf das Problem zugeschnittene Verteilung $p^*(\mathbf{x})$ zu bestimmen:

- s_{Start} : Die Startposition des zu stabilisierenden Körpers
- s_{Ende} : Die Position, die der zu stabilisierende Körper am Ende einer Simulation erreicht hat
- $s_{Bew.} := \|s_{Ende} - s_{Start}\|_2$: Die Strecke, die der Körper während der Simulation zurücklegt

Da die zurückgelegte Bewegung des Testkörpers ein naheliegendes Maß für dessen Stabilität im Raum darstellt, definieren wir $p^*(\mathbf{x}) := \frac{1}{s_{Bew.}(\mathbf{x})}$. Damit ergibt sich nach Gleichung 1.1 (Seite 6)

$$p(\mathbf{x}) = \frac{1}{Z \cdot s_{Bew.}(\mathbf{x})}, \quad (4.1)$$

wobei Z die uns unbekannte Normierungskonstante ist.

Es stellt sich nun jedoch als schwierig heraus, ein geeignetes M zur Erfüllung der Forderung 1.5 zu finden, da im besten Fall $s_{Bew.} = 0$ gilt, wobei $p^*(\mathbf{x})$ gegen unendlich geht. Um dieses Problem zu umgehen, führen wir eine Korrekturkonstante ζ ein, mit der sich

$$p^*(\mathbf{x}) = \frac{1}{s_{Bew.} + \zeta} \quad (4.2)$$

ergibt. So können wir gemäß 1.5 direkt die Formel

$$M \geq \frac{1}{\zeta} \geq \frac{1}{s_{Bew.} + \zeta} \quad (4.3)$$

für die Bestimmung von M herleiten. Setzen wir nun $\zeta = 1$, so folgt $M \geq 1$. Wir können also festlegen, dass mit $M = 1$ für $\zeta = 1$ das Kriterium 1.5 erfüllt wird und erhalten so

$$p^*(\mathbf{x}) = \frac{1}{s_{Bew.} + 1} \quad (4.4)$$

$$q(\mathbf{y}|\mathbf{x}) = p^*(\mathbf{x}). \quad (4.5)$$

Somit sind alle Größen definiert, die für das Rejection Sampling benötigt werden und wir fahren mit dem Algorithmus fort.

4.2. Algorithmus

Dieser Algorithmus ist dem in Kapitel 1.2 in großen Teilen ähnlich und unterscheidet sich nur in einigen Modifikationen.

So wurde beispielsweise die Abbruchbedingung in die innere Schleife eingebaut für den Fall, dass der zu stabilisierende Körper eine größere Strecke zurücklegt, als er dies im freien Fall tun würde. Auf diese Weise schließen wir einerseits Proben, die von der gesuchten Stabilität so weit entfernt sind, dass wir sie nicht berücksichtigen wollen, generell aus und schränken unsere Verteilung auf geeignetere Werte ein, andererseits minimieren wir so die durch die weiter oben aufgeführten Probleme der Simulationsumgebung auftretende Verfälschung der Versuchswerte.

Außerdem wurden die Schritte berücksichtigt, die zur Berechnung von $q(\mathbf{y}|\mathbf{x})$ nötig sind. Der Pseudocode zum kompletten Algorithmus findet sich in Algorithm 3.

4.3. Anwendung auf Problem 1

4.3.1. Versuchsparameter

Um den Nutzen des Rejection Sampling für die vorliegende Problemstellung zuverlässig bewerten zu können, werden mehrere Versuche mit unterschiedlichen Parametern bezüglich der Anzahl der zur Stabilisierung simulierten Körper und der Höhe des zu stabilisierenden Körpers durchgeführt. Parameter, die in allen Versuchen den selben Wert haben sind:

- Anzahl gezogener Proben: 100.000
- Raum, in dem die zufälligen Körper simuliert werden: $x \in [-4, 4]$; $y \in [-4, 4]$; $z \in [0, \textit{toleranz}]$

Algorithm 3 Rejection Sampling in der Simulationsumgebung

```
1: Anzahl  $n_{Körper}$  der zu simulierenden Körper, Anzahl  $n_{Probe}$  der durchzuführenden
   Versuche, Höhe  $h$  des zu stabilisierenden Körpers und Toleranz für das Abbruch-
   kriterium festlegen
2: for  $k = 1$  bis  $n_{Probe}$  do
3:   repeat
4:     Zu stabilisierenden Körper und  $n_{Körper}$  andere zufällig verteilte Körper ge-
     nerieren  $\hat{=}$  Probe  $x^{Probe}$  aus  $q(x)$  ziehen
5:     Anfangsposition  $s_{Start}$  des zu stabilisierenden Körpers erfassen
6:     Simulation für Zeit  $t$ 
7:     Endposition  $s_{Ende}$  des zu stabilisierenden Körpers erfassen
8:     Sei  $s_{Bew.} = \|s_{Ende} - s_{Start}\|_2$ 
9:     if  $s_{Bew.} \geq Toleranz$  then go to 3
10:     $a = \frac{1}{s_{Bew.} + 1}$ 
11:     $u$  gleichverteilt aus  $[0, 1]$  ziehen
12:   until  $u \leq a$ 
13:    $x^l = x^{Probe}$ 
```

Tabelle 4.1.: Rejection Sampling - Versuchsparameter

Versuch	Anzahl Körper	Höhe H
1	8	2
2	10	2,5
3	12	3
4	14	3,5
5	16	4
6	18	4,5
7	20	5

Die bei den einzelnen Versuchen verwendeten Parameter sind in Tabelle 4.1 aufgelistet.

Hierbei ist zu beachten, dass die Anzahl der zufällig simulierten Körper mit den Abmessungen ($1 \times 1 \times 0.25$) der Höhe des zu stabilisierenden Körpers dergestalt angepasst wurde, dass exakt so viele Körper vorhanden sind, wie zur perfekten Stabilisierung notwendig sind.

4.3.2. Ergebnisanalyse

Die Ergebnisse der Versuchsreihe finden sich in der Tabelle 4.2. Werte, die wir im Rahmen der Analyse betrachten wollen, sind der beste Stabilitätswert a_{max} , die mittlere Stabilität \bar{a} , die minimale beobachtete Bewegung $s_{Bew.min}$ und die mittlere Bewegung

Tabelle 4.2.: Rejection Sampling in Problem 1 mit 100.000 Proben

Versuch	a_{max}	\bar{a}	$s_{Bew.min}$	$\bar{s}_{Bew.}$	$n_{Abbr.}$
1	0,265	0,020	0,879	1,659	431.465
2	0,168	0,014	1,360	2,151	403.608
3	0,116	0,011	1,844	2,640	403.434
4	0,093	0,009	2,264	3,132	383.876
5	0,064	0,007	2,830	3,625	348.509
6	0,051	0,006	3,321	4,116	350.206
7	0,049	0,006	3,660	4,606	341.026

$\bar{s}_{Bew.}$ des zu stabilisierenden Körpers, sowie die mittlere Anzahl der durch das Toleranzkriterium verursachten Abbrüche $n_{Abbr.}$.

Da die in den Versuchen ermittelten Werte a definitionsgemäß eine Untergrenze abweichend von 0 haben, wurden diese gemäß der Formel $a_{neu} = \frac{a_{alt} - a_{min}}{1 - a_{min}}$ an das Intervall $[0, 1]$ angepasst.

Es ist direkt ersichtlich, dass Rejection Sampling für das vorliegende Problem ungeeignet ist. Dies ist einerseits an der im Verhältnis zu der Menge der Versuche sehr hohen Anzahl an Abbrüchen zu sehen; andererseits an den kleinen Werten von maximaler und mittlerer Stabilität a_{max} und \bar{a} .

4.3.3. Visualisierung der Versuchsergebnisse

Die Einbeziehung der für uns relevanten Parameter Position $\in \mathbb{R}^3$ und Rotation um die z-Achse $\in \mathbb{R}$ des Körpers führt dazu, dass wir bei einem Versuch mit n Körpern einen Zustandsraum der Dimension $4 \cdot n$ betrachten. Dementsprechend gestaltet sich die Visualisierung der Versuchsergebnisse als schwierig.

In Abbildung 4.1 (Seite 27) sind die Verläufe der Stabilitätswerte über den Nummern der jeweiligen Versuche aufgetragen. Wie zu erwarten war, sind die Verläufe chaotisch, da die einzelnen Versuche in keiner Korrelation zueinander stehen.

Abbildung 4.2 (Seite 28) zeigt die Schaubilder der Häufigkeit der auftretenden Stabilitätswerte, wobei diese in Intervalle der Größe 0,001 aufgeteilt wurden. Der häufigste Stabilitätswert, sowie dessen Häufigkeit, sind dabei jeweils in den Schaubildern angegeben. Hier ist wieder erkenntlich, dass die simulierten Konstellationen überwiegend instabil sind.

Einige repräsentative Versuchsergebnisse, die den häufigsten Stabilitätswerten entsprechen, sind in Abbildung 4.3 (Seite 29) zu sehen. Diese Stabilitäten ergeben sich in den meisten Fällen, wenn der zu stabilisierende Körper durch nur einen weiteren Quader gestützt wird.

Repräsentationen der stabilsten Konstellationen finden sich in Abbildung 4.4 (Seite 30). Es ist zu beachten, dass einige dieser Konstellationen nicht so stabil sind, wie dies durch den Stabilitätswert suggeriert wird, da der zu stabilisierende Körper

Tabelle 4.3.: Rejection Sampling Problem 2 - Versuchsparameter

Versuch	Anzahl Körper	H	b
1	8	1,6	2
2	10	2	2,5
3	12	2,4	3
4	14	2,8	3,5
5	16	3,2	4
6	18	3,6	4,5
7	20	4	5

beispielsweise hochkant stehen oder durch einen senkrecht stehenden Quader stabilisiert werden kann. Ein hoher Wert der Stabilität kann folglich nicht als letztendlicher Garant für tatsächliche Stabilität gelten.

4.4. Anwendung auf Problem 2

4.4.1. Versuchsparameter

Wie auch schon in Problemstellung 1 werden hier die Parameter variiert, um die Güte unseres Rejection Sampling Algorithmus beurteilen zu können. Einige Parameter bleiben jedoch auch hier in allen Versuchen gleich:

- Anzahl gezogener Proben: 100.000
- Höhe der Ufer: $H_2 = 5$

Variiert wird hier sowohl die Höhe des zu stabilisierenden Körpers, als auch der Abstand der beiden Ufer. Der Raum, in dem zufällige Körper generiert werden, wird dem entsprechend angepasst. Die Bezeichnungen der einzelnen Parameter sind Abbildung 3.4 zu entnehmen.

Da die in *Problem 1* verwendete Funktion $p^*(x) = \frac{1}{s_{Bew.}+1}$ hier kein sinnvolles Maß zur Stabilität der Konstellation darstellt, führen wir hier

$$p^*(x) = \begin{cases} 1, & \text{falls } s_{Bew.} \leq \textit{toleranz} \\ 0, & \text{sonst} \end{cases} \quad (4.6)$$

als neue Funktion ein.

Die Abmessungen der zur Stabilisierung generierten Körper, sowie des zu stabilisierenden Körpers betragen in diesem Fall $(1 \times 1 \times 0, 1)$. Die Körper werden jeweils, wie in Abbildung 4.7 (Seite 33) zu sehen, symmetrisch generiert, um so die Wahrscheinlichkeit für die Entstehung einer Brücke zu erhöhen.

Tabelle 4.4.: Rejection Sampling in Problem 2 mit 100.000 gezogenen Proben

Versuch	Anzahl stabiler Konstellationen
1	9
2	1
3	1
4	0
5	1
6	3
7	0

4.4.2. Ergebnisanalyse

Die Ergebnisse der Versuchsreihe sind in Tabelle 4.4 aufgelistet.

Es ist zu sehen, dass Rejection Sampling nicht sehr effizient ist, jedoch einige stabile Konstellationen liefert.

4.4.3. Visualisierung der Versuchsergebnisse

Der Verlauf der Stabilität über der Zeit, sowie deren Häufigkeit sind in den Abbildungen 4.5 (Seite 31) und 4.6 (Seite 32) zu sehen. Wie zu erwarten war, liegt dem Verlauf keinerlei Ordnung zu Grunde, da die einzelnen Proben nicht miteinander korreliert sind. Des Weiteren lässt sich aus beiden Abbildungen ersehen, dass der überwiegende Teil der Proben instabil ist.

In Abbildung 4.7 (Seite 33) ist eine der stabilen Konstellationen abgebildet.

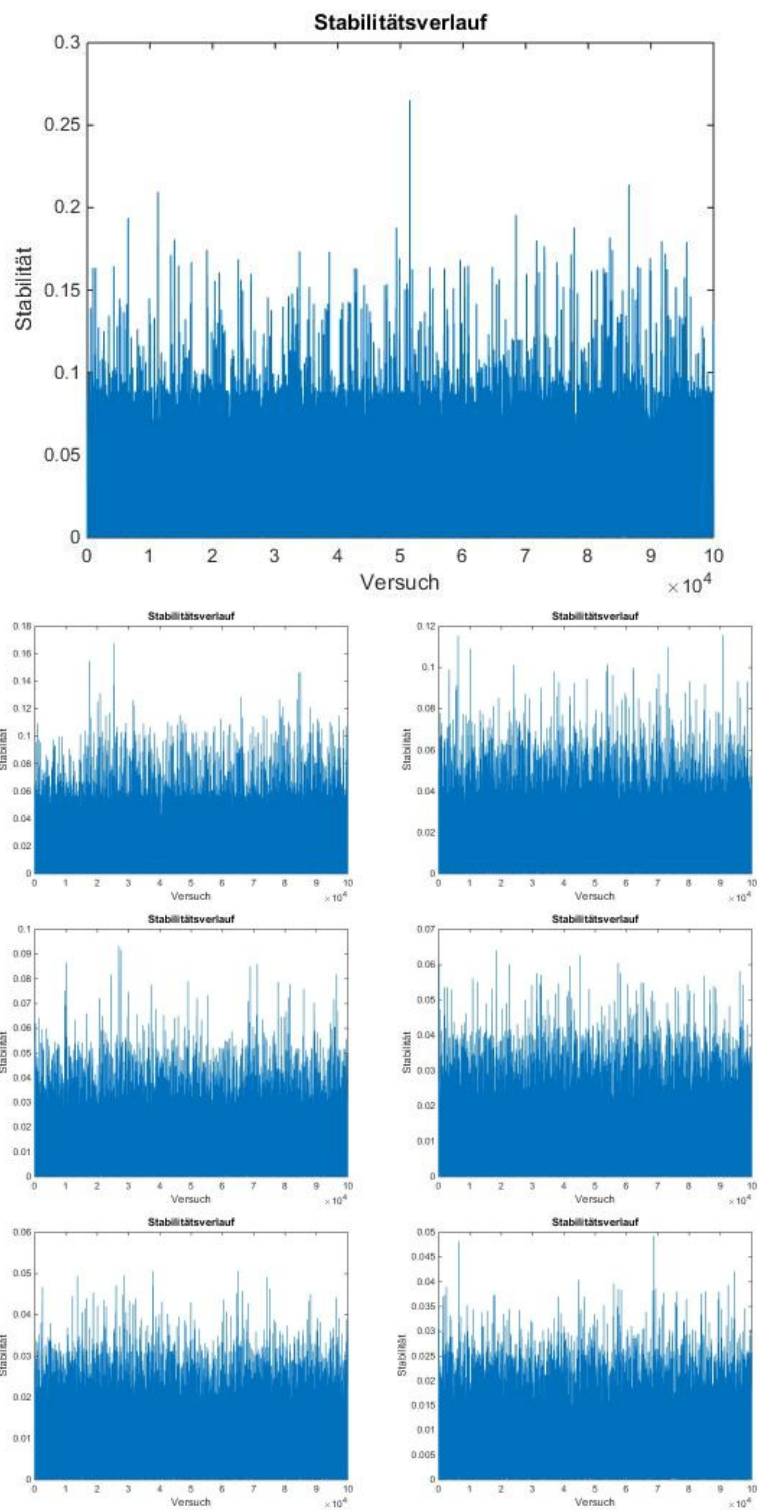


Abbildung 4.1.: Zeitlicher Verlauf der Stabilitätswerte

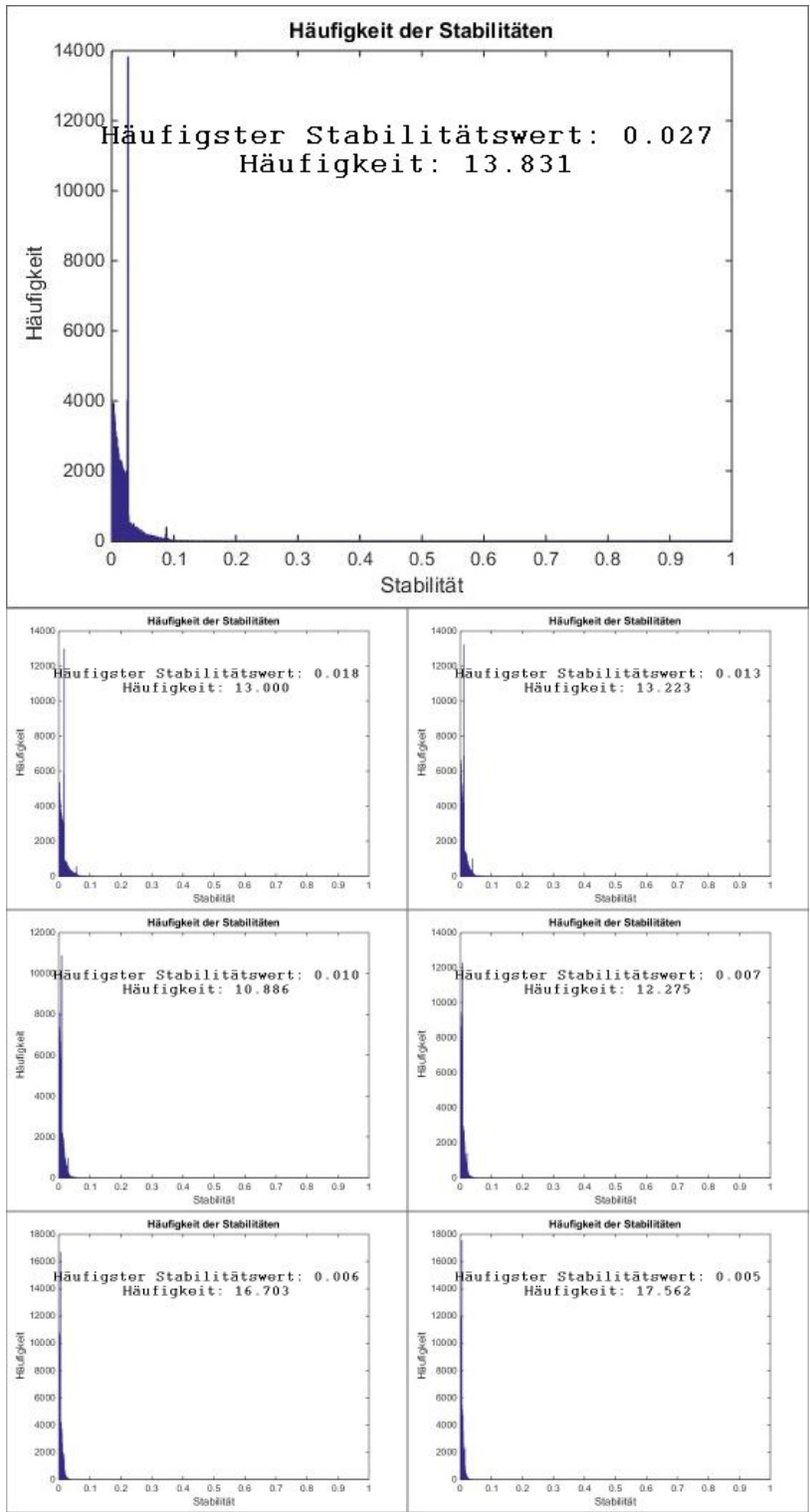


Abbildung 4.2.: Häufigkeit der verschiedenen Stabilitätswerte

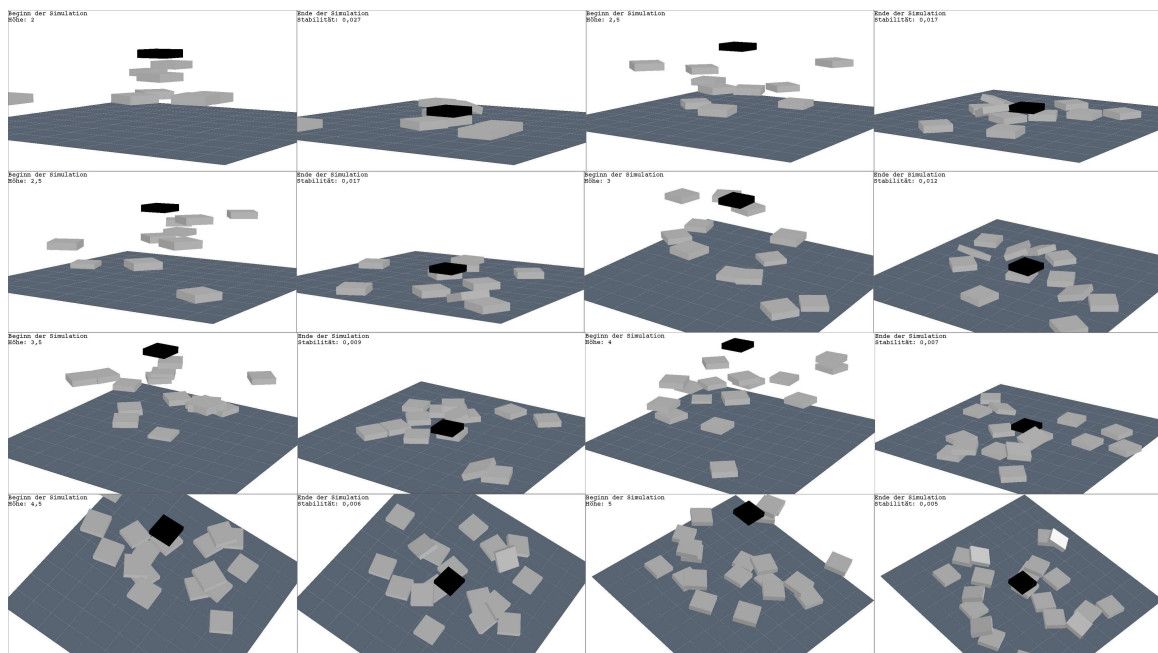


Abbildung 4.3.: Auswahl der Konstellationen mit den häufigsten Stabilitäten

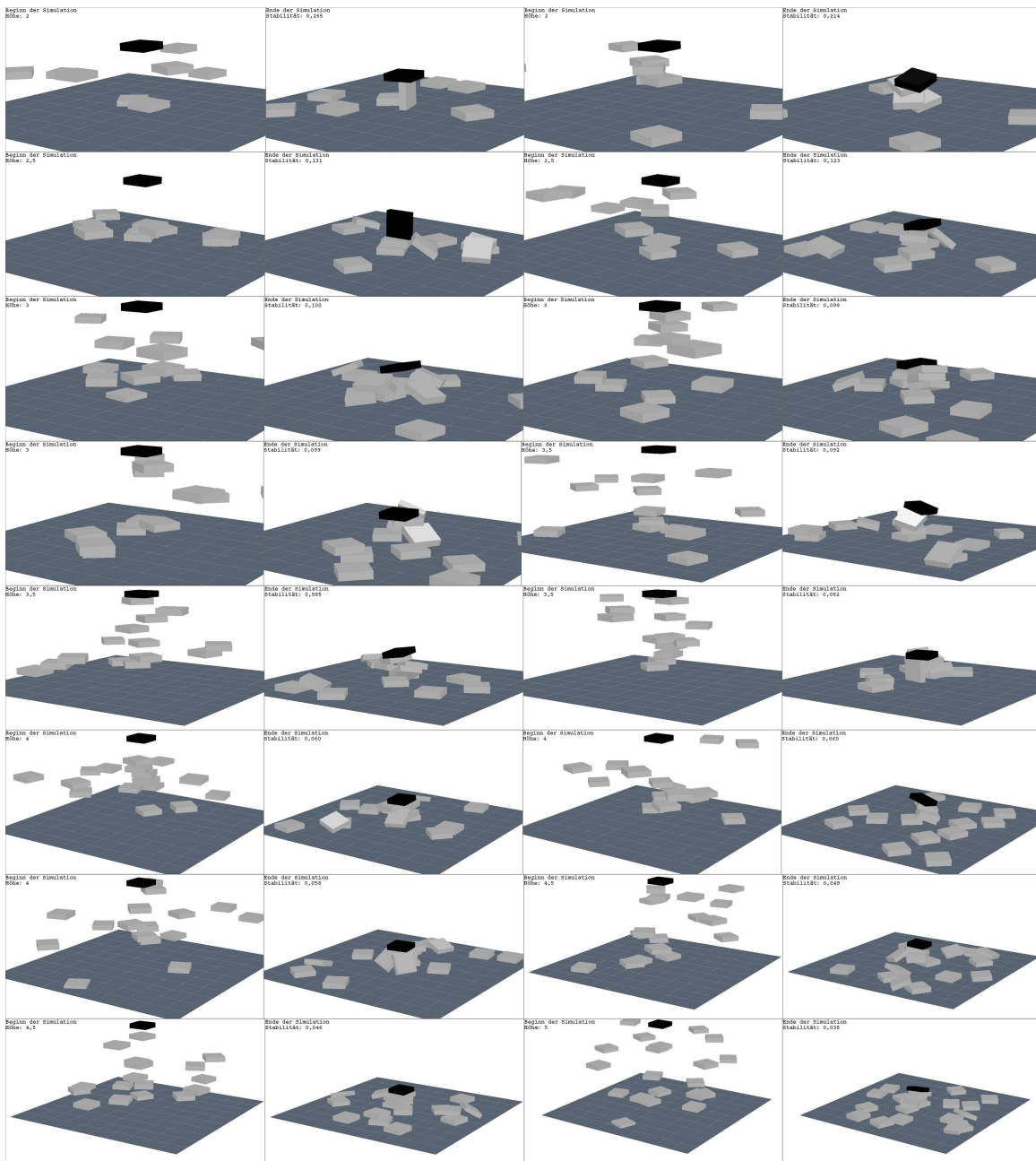


Abbildung 4.4.: Auswahl der besten Konstellationen aus Rejection Sampling

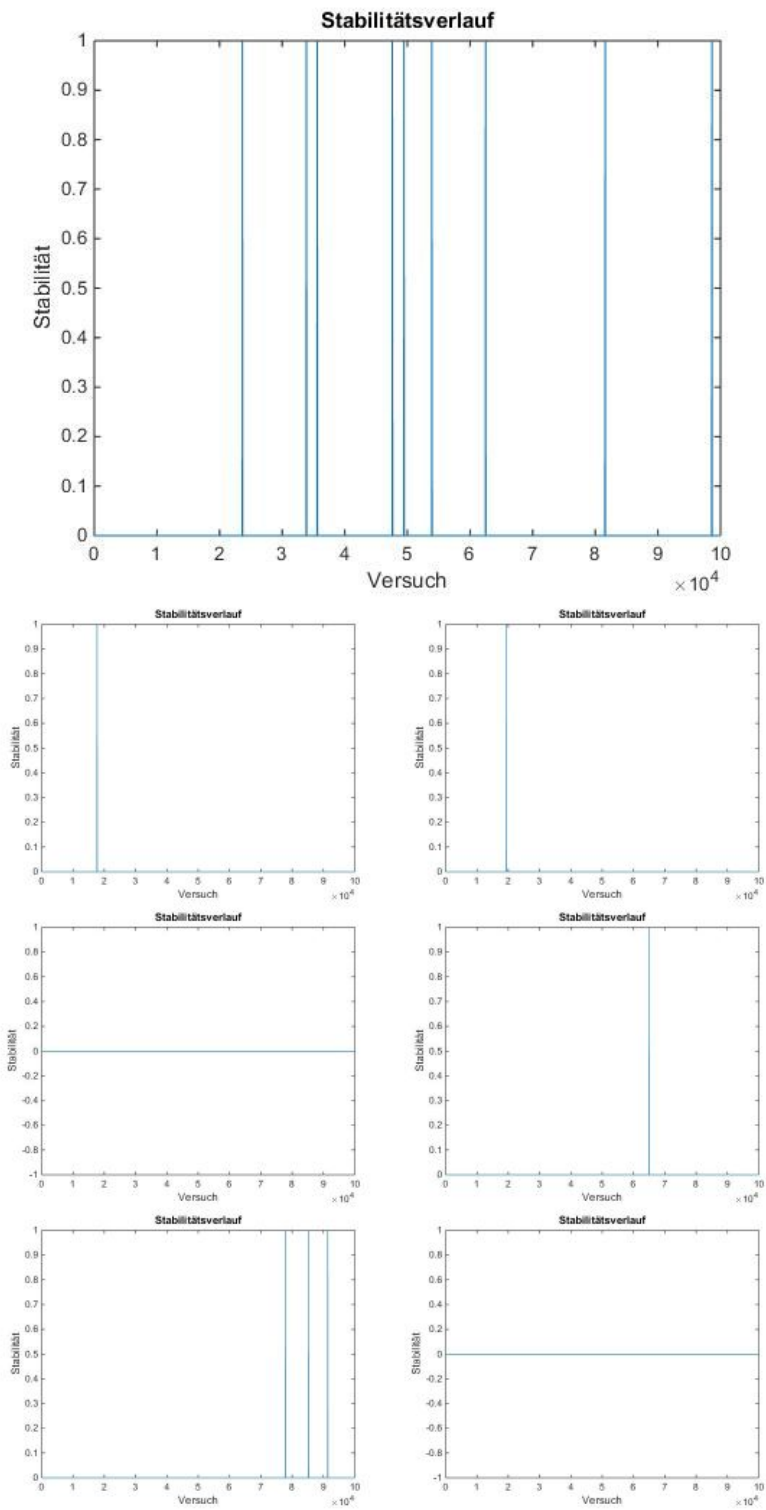


Abbildung 4.5.: Verlauf der Stabilität über der Zeit in Rejection Sampling

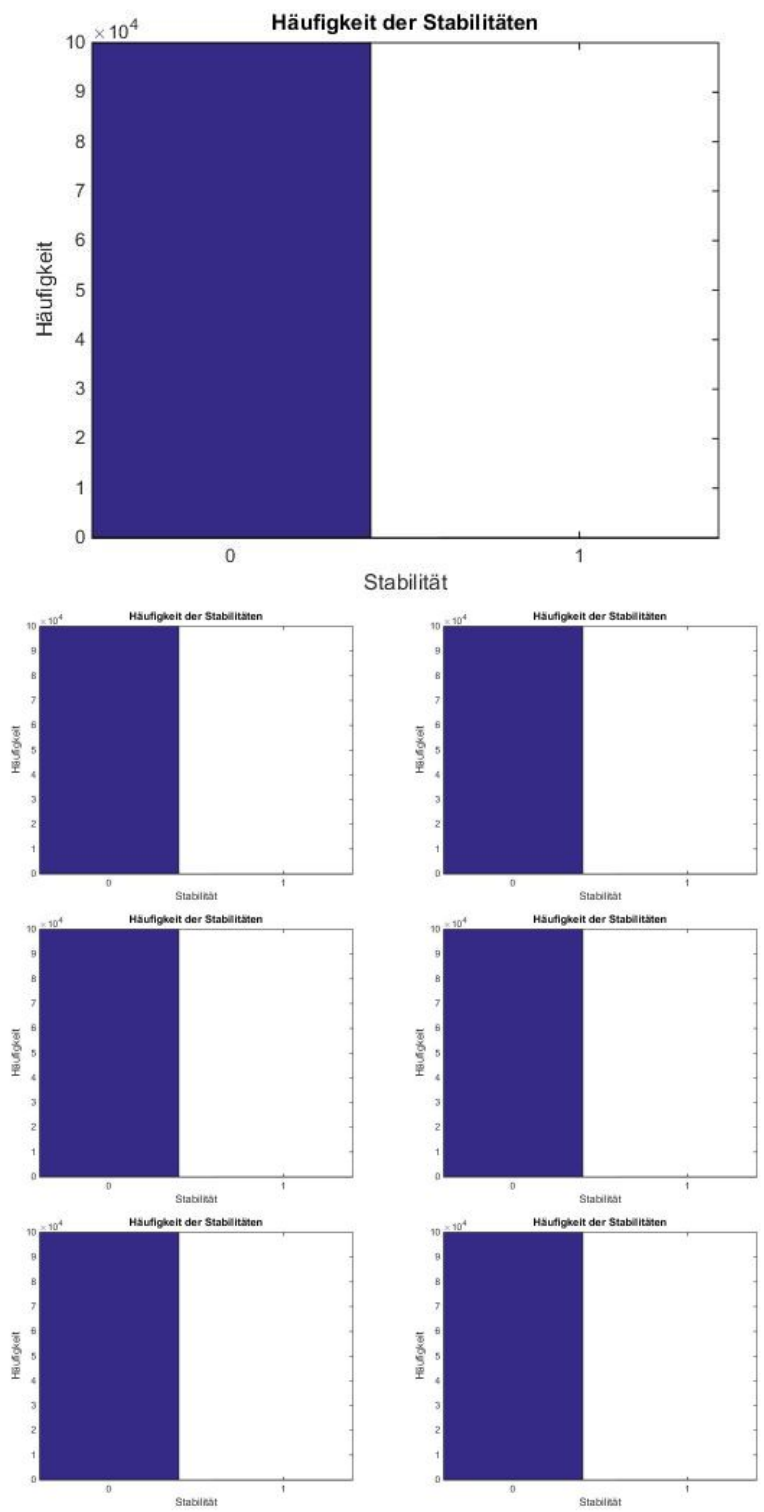


Abbildung 4.6.: Verteilung der Stabilitätswerte in Rejection Sampling

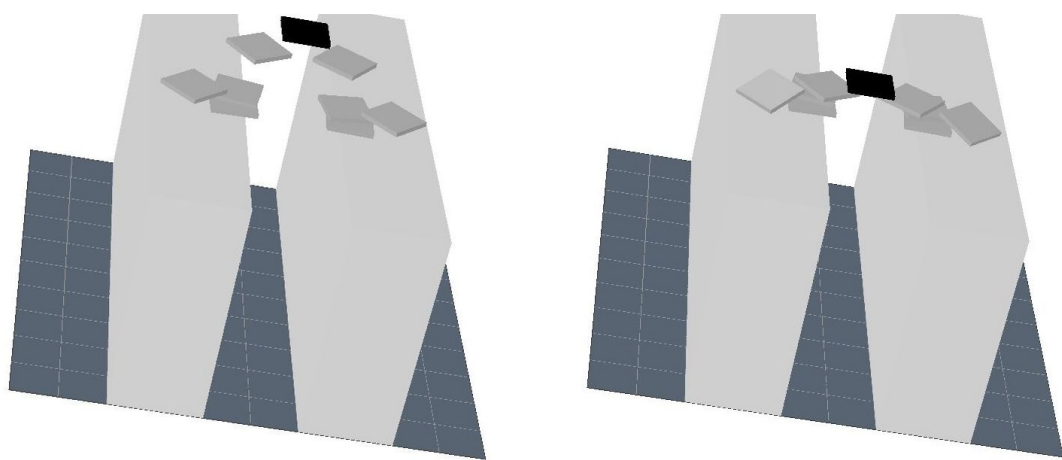


Abbildung 4.7.: Stabile Konstellation mit $b = 2$

5. MCMC-Verfahren

5.1. Problemstellung in MCMC

Einen Teil der auf die MCMC-Methode abgestimmten Definitionen für unser Problem können wir direkt aus Kapitel 4.1 übernehmen. So bietet die zurückgelegte Strecke des zu stabilisierenden Körpers $s_{Bew.}$ auch hier eine gute Möglichkeit, die Stabilität abzuschätzen. Daher erscheint es ratsam, für $p^*(\mathbf{x})$ die selbe Funktion wie beim Rejection Sampling zu wählen:

$$p^*(\mathbf{x}) = \frac{1}{s_{Bew.} + 1} \quad (5.1)$$

Bei der MCMC-Methode muss allerdings noch $\tilde{q}(x'|x)$ aus dem Markov-Übergang 2.7 definiert werden. Hier bietet sich die Gauß'sche Normalverteilung an, da man aus dieser effizient Proben ziehen kann und weil sich, wie wir gleich sehen werden, Vereinfachungen für unsere Funktion $f(\mathbf{x}', \mathbf{x})$ ergeben. Wir legen also fest:

$$\tilde{q}(x'|x) = \mathcal{N}(\mathbf{x}'|x, \sigma^2 \mathbf{I}) \propto e^{-\frac{1}{2\sigma^2}(x'-x)^2}. \quad (5.2)$$

Aus $(x' - x)^2 = (x - x')^2$ folgt direkt $\mathcal{N}(\mathbf{x}'|x, \sigma^2 \mathbf{I}) = \mathcal{N}(\mathbf{x}|x', \sigma^2 \mathbf{I})$ und somit $\tilde{q}(x'|x) = \tilde{q}(x|x')$. Dadurch vereinfacht sich unsere Funktion $f(\mathbf{x}', \mathbf{x})$ gemäß 2.18 zu

$$f(\mathbf{x}', \mathbf{x}) = \min \left(1, \frac{p^*(\mathbf{x}')}{p^*(\mathbf{x})} \right). \quad (5.3)$$

Somit sind alle Voraussetzungen für die MCMC-Methode festgelegt und wir können im folgenden Abschnitt mit dem eigentlichen Algorithmus fortfahren.

5.2. Algorithmus

Dieser Algorithmus unterscheidet sich von Algorithm 2 lediglich in einigen kleinen Modifikationen. So wurden wie schon bei Rejection Sampling die Schritte zur Berechnung von $p^*(\mathbf{x}) = \frac{1}{s_{Bew.} + 1}$ eingefügt, sowie eine Abbruchbedingung für Proben außerhalb jeglicher Stabilität festgelegt.

Algorithm 4 MCMC in der Simulationsumgebung

```
1: Anzahl Körper, Anzahl Proben  $n_{Probe}$  und Toleranz festlegen, Startprobe  $x^1$  simulieren
2: Zu stabilisierenden Körper generieren
3: for  $l = 2$  bis  $L$  do
4:   Probe  $x^l$  aus  $\mathcal{N}(\mathbf{x}^l|x^{l-1}, \sigma^2\mathbf{I})$  ziehen
5:    $s_{Start}^l$  speichern und Simulation starten
6:    $s_{Ende}^l$  speichern
7:    $s_{Bew.}^l = \|s_{Ende}^l - s_{Start}^l\|_2$ 
8:   if  $s_{Bew.}^l \geq H + H_2$  then go to 4
9:    $p^*(\mathbf{x}^l) = \frac{1}{s_{Bew.}^l + 1}$ 
10:  Setze  $a = \frac{p^*(\mathbf{x}^l)}{p^*(\mathbf{x}^{l-1})}$ 
11:  if  $a \geq 1$  then  $x^l$  wird als Probe akzeptiert
12:  else
13:    Ziehe gleichverteilt ein  $u \in [0, 1]$ 
14:    if  $u \leq a$  then  $x^l$  wird als Probe akzeptiert
15:  else
16:    Setze  $x^l = x^{l-1}$ 
```

5.3. Anwendung auf Problem 1

5.3.1. Versuchsparameter

Für die MCMC-Methode verwenden wir die selben Parameter, wie schon beim Rejection Sampling. Das heißt 100.000 gezogene Proben je Versuch und simulierte Körper in dem Gebiet $x \in [-4, 4], y \in [-4, 4], z \in [0, \textit{toleranz}]$. Auch haben die zur Stabilisierung simulierten Körper die selben Abmessung wie im Rejection Sampling zu *Problem 1*.

Im Gegensatz zum Rejection Sampling muss hier jedoch noch σ in $\mathcal{N}(\mathbf{x}^l|x^{l-1}, \sigma^2\mathbf{I})$ festgelegt werden. Wir bestimmen für die verschiedenen Koordinaten unterschiedliche σ . Einerseits für die Position $\sigma_{Pos.}$ der aus der Übergangsverteilung gezogenen Körper, andererseits für deren Rotation $\sigma_{Rot.}$.

Wir führen an dieser Stelle zwei Versuchsreihen mit unterschiedlichen Werten für σ durch:

- Messreihe 1: $\sigma_{Pos.} = 0,5$ und $\sigma_{Rot.} = \pi/8$
- Messreihe 2: $\sigma_{Pos.} = 1$ und $\sigma_{Rot.} = \pi/4$

Diese Parameterwerte werden wir auch später in den modifizierten Varianten der MCMC-Methode zu *Problem 1* verwenden. Die übrigen Parameter sind die selben wie im Rejection Sampling zu diesem Problem und in Tabelle 5.1 zu sehen.

Tabelle 5.1.: MCMC - Versuchsparameter

Versuch	Anzahl Körper	Höhe
1	8	2
2	10	2,5
3	12	3
4	14	3,5
5	16	4
6	18	4,5
7	20	5

Tabelle 5.2.: MCMC in Problem 1 mit 100.000 Proben, $\sigma = 0,5$ und $\sigma_{rot.} = \pi/8$

Versuch	a_{max}	\bar{a}	$s_{Bew.min}$	$\bar{s}_{Bew.}$	$n_{Abbr.}$
1	0,265	0,024	0,879	1,641	35.703
2	0,226	0,033	1,197	2,133	34.115
3	0,137	0,025	1,778	2,622	33.463
4	0,095	0,019	2,309	3,114	32.245
5	0,065	0,009	2,814	3,605	29.255
6	0,082	0,007	2,895	4,095	27.966
7	0,049	0,006	3,670	4,584	27.051

5.3.2. Ergebnisanalyse

Die Ergebnisse zu beiden Versuchsreihen sind in den Tabellen 5.2 und 5.3 aufgelistet.

Wie direkt zu sehen ist, sind die Ergebnisse nur marginal besser als im Rejection Sampling. Die Werte a_{max} und \bar{a} sind hier in den meisten Fällen ein wenig größer und in einigen Fällen gleich groß, jedoch nirgends kleiner. Eine letztendlich stabile Konstellation wird jedoch nicht gefunden.

Die zweite Versuchsreihe unterscheidet sich von der ersten hauptsächlich in der Anzahl der Abbrüche, die etwa doppelt so hoch ist. Dies rührt möglicherweise daher, dass die einzelnen Proben aufgrund der größeren σ -Werte weniger stark korreliert sind und die Körper in den einzelnen Proben daher mit höherer Wahrscheinlichkeit weiter auseinander liegen, wie dies schon im Rejection Sampling der Fall war. Bezüglich der Stabilitätswerte sind sich die beiden Reihen sehr ähnlich und es lässt sich aus den Ergebnissen nicht ableiten, welche σ -Werte besser geeignet sind, um eine stabile Konstellation zu finden.

Im Großen und Ganzen ist zu sehen, dass die MCMC-Methode, wie sie hier verwendet wird, das Problem nicht zufriedenstellend löst, jedoch etwas bessere Ergebnisse als Rejection Sampling erzielt.

Tabelle 5.3.: MCMC in Problem 1 mit 100.000 Proben, $\sigma = 1$ und $\sigma_{rot.} = \pi/4$

Versuch	a_{max}	\bar{a}	$s_{Bew.min}$	$\bar{s}_{Bew.}$	$n_{Abbr.}$
1	0,301	0,024	0,802	1,641	69.828
2	0,209	0,018	1,212	2,129	34.115
3	0,131	0,014	1,759	2,618	53.388
4	0,110	0,011	2,134	3,107	47.484
5	0,100	0,009	2,552	3,599	39.619
6	0,059	0,008	3,197	4,085	35.968
7	0,056	0,008	3,569	4,573	32.735

5.3.3. Visualisierung der Versuchsergebnisse

Wie in den Abbildungen 5.1 (Seite 40), 5.2 (Seite 41), 5.3 (Seite 42) und 5.4 (Seite 43) zu sehen ist, sind die Versuchsergebnisse jenen des Rejection Sampling sehr ähnlich.

Überraschend ist, dass der zeitliche Verlauf der Wahrscheinlichkeitswerte noch immer sehr chaotisch ist, obwohl aufeinanderfolgende Proben korreliert sind und man infolge der Begünstigung von Proben mit höherer Stabilität einen Aufwärtstrend erwarten würde. Dieser Sachverhalt weist darauf hin, dass instabile Konstellationen um ein vielfaches wahrscheinlicher sind, als stabile Konstellationen.

In den Abbildungen 5.3 und 5.4 ist eine Auswahl von Konstellationen zu sehen, die in den Versuchen die höchste Stabilität, beziehungsweise die am häufigsten auftretenden Stabilitäten besitzen.

5.4. Anwendung auf Problem 2

Da hier, wie schon in der Anwendung des Rejection Sampling auf *Problem 2*, $p^*(x) = \frac{1}{s_{Bew}+1}$ kein zuverlässiges Maß für die Stabilität der jeweiligen Konstellation darstellt, muss auch hier eine andere Funktion verwendet werden, die in diesem Fall für die MCMC-Methode geeignet ist.

In der ersten Versuchsreihe wählen wir

$$p^*(x^l) = u(l) \cdot p^*(x^{l-1}), \quad (5.4)$$

wobei $u(l)$ einen zufälligen, gleichverteilten Wert aus dem Intervall $[0, 1]$ annimmt. $p^*(x^1)$ wird dabei nach der Formel $p^*(x^1) = \frac{1}{s_{Bew}+1}$ bestimmt.

Somit ergibt sich $a = \frac{p^*(x^l)}{p^*(x^{l-1})} = u$. Die MCMC-Methode wird hierdurch dem Rejection Sampling ähnlich. Daher erwarten wir hier ähnliche Versuchsergebnisse.

Tabelle 5.4.: MCMC Problem 2 - Versuchsparameter

Versuch	Anzahl Körper	H	b
1	8	1,6	2
2	10	2	2,5
3	12	2,4	3
4	14	2,8	3,5
5	16	3,2	4
6	18	3,6	4,5
7	20	4	5

Tabelle 5.5.: MCMC in Problem 2 mit $\sigma_{Pos.} = 0,5$ und $\sigma_{Rot.} = \pi/8$

Versuch	Anzahl stabiler Konstellationen	Anzahl voneinander verschiedener stabiler Konstellationen
1	5	1
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0

5.4.1. Versuchsparameter

Als Parameter werden die selben gewählt, die bei der Anwendung des Rejection Samplings auf *Problem 2* verwendet wurden.

Auch hier gilt es, wie bei der Anwendung der MCMC-Methode auf *Problem 1*, σ festzulegen. Wir führen an dieser Stellen wieder zwei Versuchsreihen zu je 100.000 Proben je Versuch durch und wählen im ersten Fall $\sigma_{Pos.} = 0,5$ und $\sigma_{Rot.} = \pi/8$; im zweiten Fall $\sigma_{Pos.} = 1$ und $\sigma_{Rot.} = \pi/4$.

Die weiteren Versuchsparameter befinden sich in Tabelle 5.4.

5.4.2. Ergebnisanalyse

Wie in den Tabellen 5.5 und 5.6 zu sehen ist, sind die Versuchsergebnisse der MCMC-Methode mit der verwendeten Funktion $p(x)$ sehr schlecht. Es wurde in insgesamt 1.400.000 Versuchen nur eine stabile Konstellationen gefunden.

Das mit dieser Methode weniger stabile Konstellationen gefunden werden, als mit Rejection Sampling, liegt daran, dass MCMC in jedem Schleifendurchlauf, der die Toleranzgrenzen nicht überschreitet, als Probe akzeptiert, während im Rejection Sampling die Proben zufällig akzeptiert werden.

Tabelle 5.6.: MCMC in Problem 2 mit $\sigma_{Pos.} = 1$ und $\sigma_{Rot.} = \pi/4$

Versuch	Anzahl stabiler Konstellationen
1	0
2	0
3	0
4	0
5	0
6	0
7	0

5.4.3. Visualisierung der Versuchsergebnisse

Auf eine Visualisierung der Versuchsergebnisse wie in den Fällen weiter oben wird an dieser Stelle aufgrund der Eindeutigkeit und Einfachheit der Versuchsergebnisse verzichtet. Die gefundene stabile Konstellation ist aus zwei Perspektiven in Abbildung 5.5 (Seite 44) zu sehen.

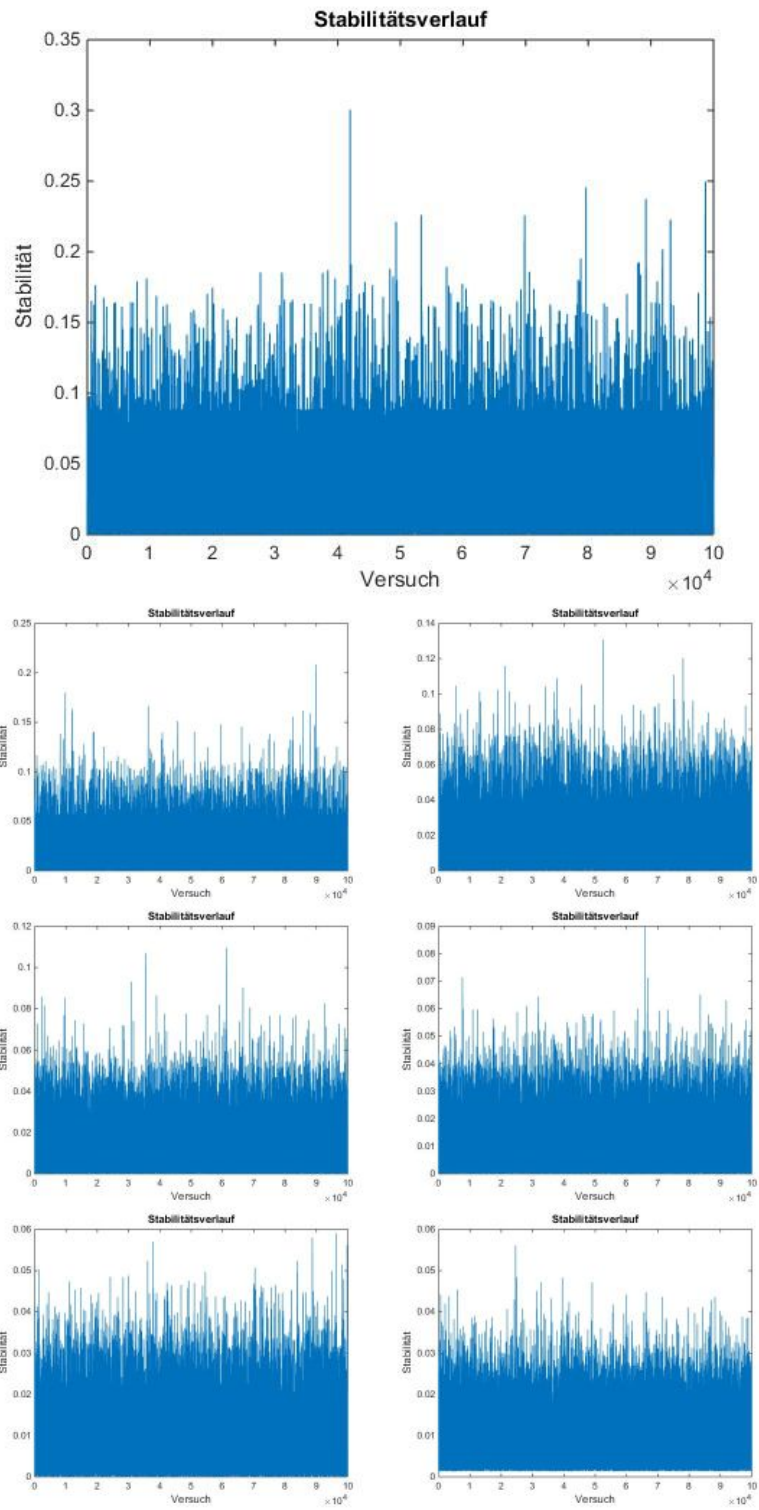


Abbildung 5.1.: Verlauf der Stabilität über der Zeit in MCMC mit $\sigma_{Pos.} = 1$ und $\sigma_{Rot.} = \pi/4$

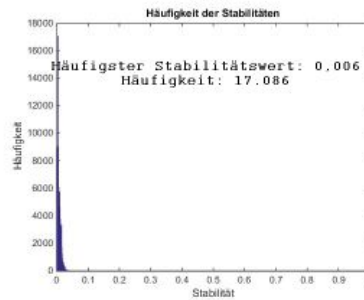
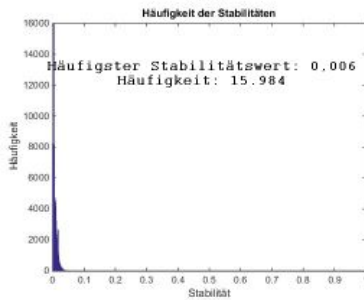
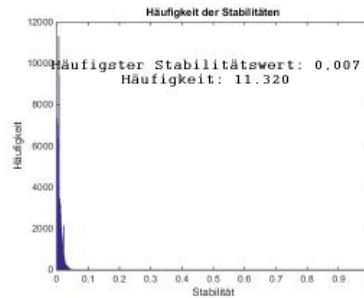
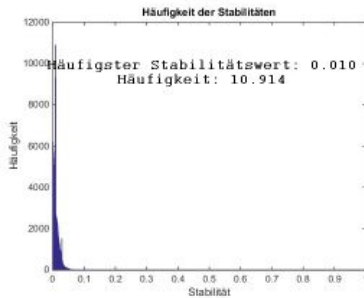
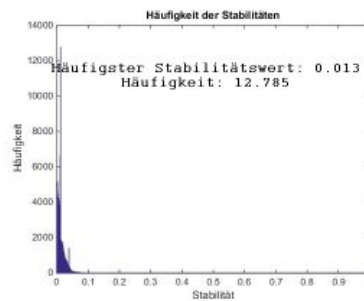
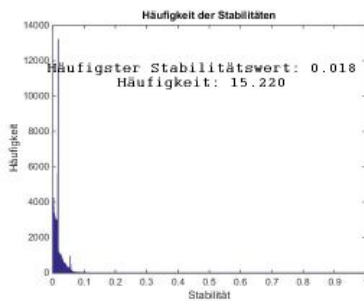
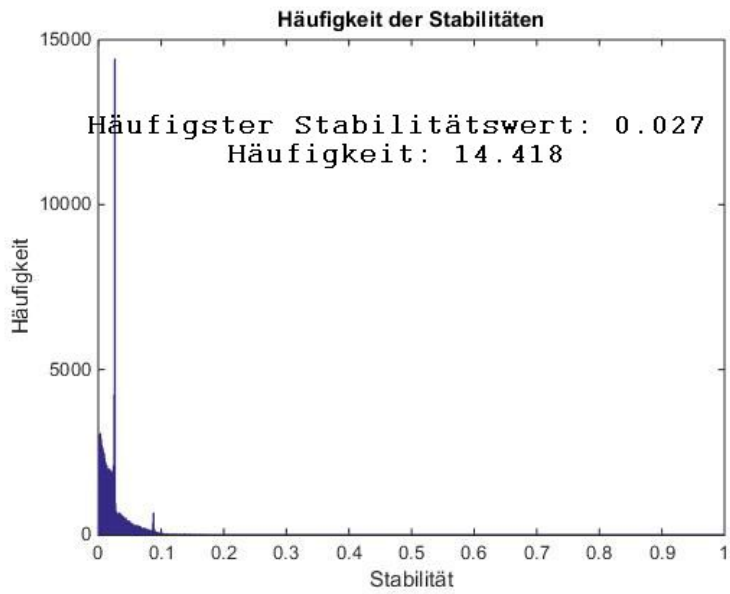


Abbildung 5.2.: Häufigkeit der Stabilitätswerte in MCMC mit $\sigma_{Pos.} = 1$ und $\sigma_{Rot.} = \pi/4$

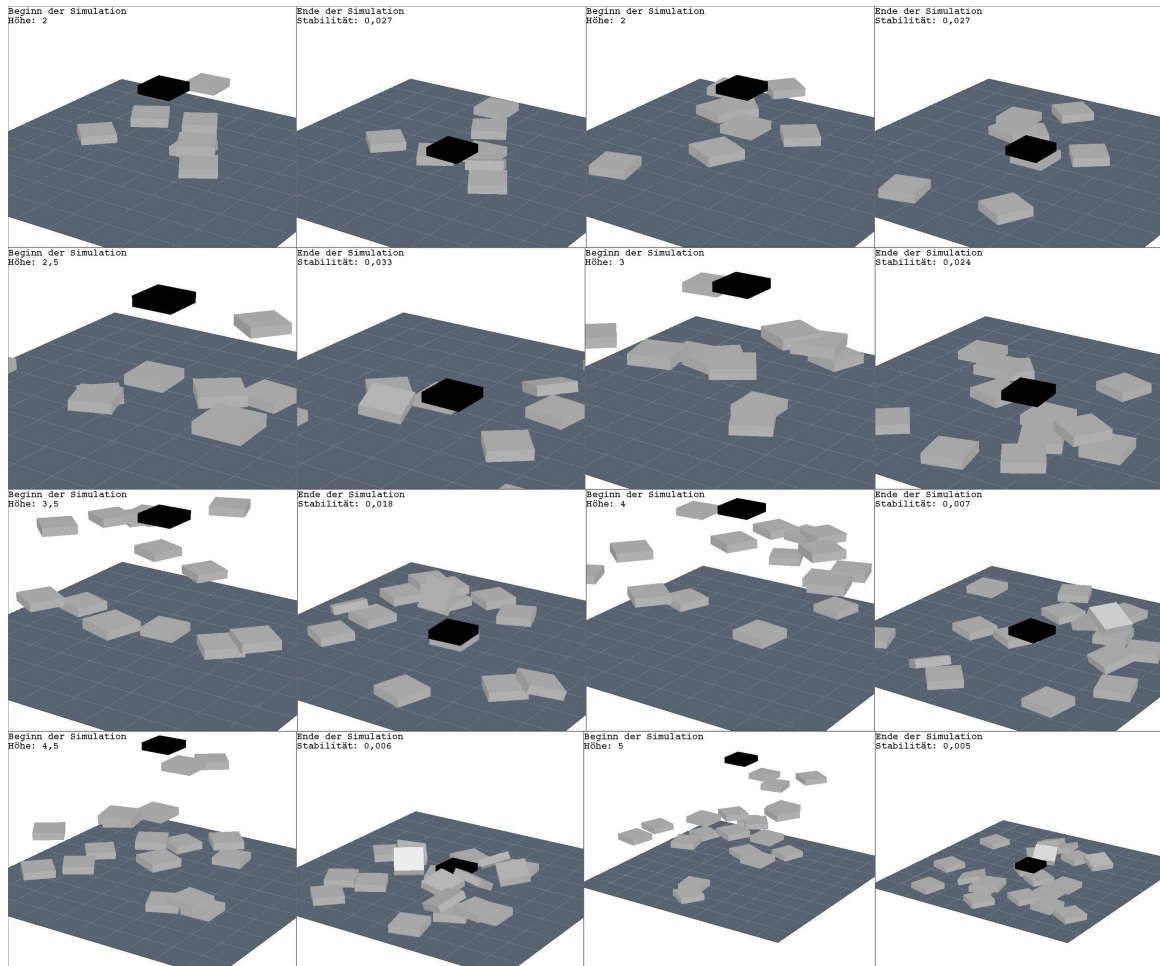


Abbildung 5.3.: Auswahl der Konstellationen mit den häufigsten Stabilitäten in MCMC mit $\sigma_{Pos.} = 0,5$ und $\sigma_{Rot.} = \pi/8$

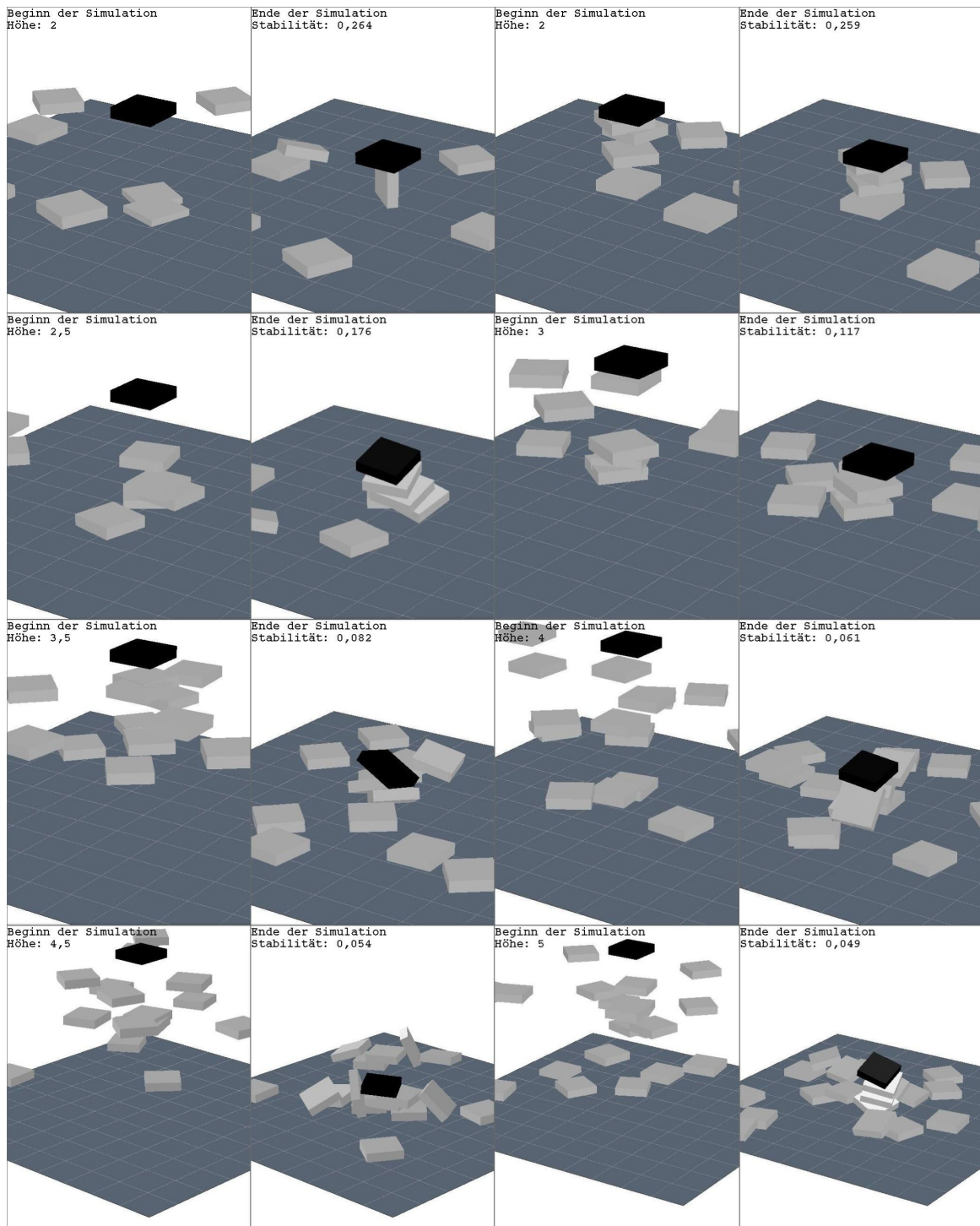
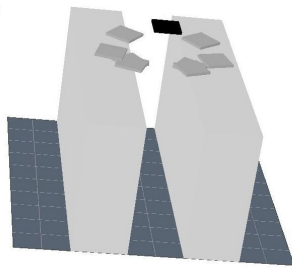
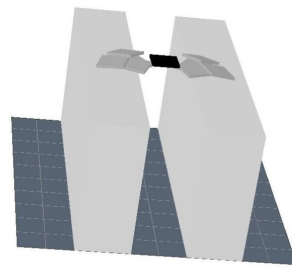


Abbildung 5.4.: Auswahl der besten Konstellationen aus MCMC mit $\sigma_{Pos.} = 0,5$ und $\sigma_{Rot.} = \pi/8$

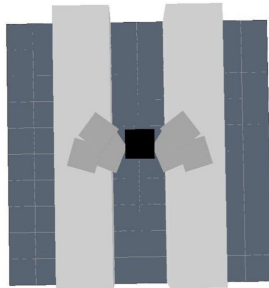
Beginn der Simulation
Abstand: 2.0



Ende der Simulation
Abstand: 2.0



Beginn der Simulation
Abstand: 2.0



Ende der Simulation
Abstand: 2.0

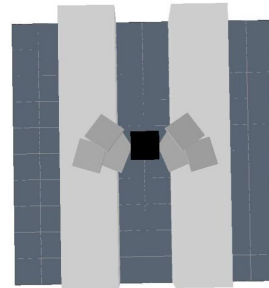


Abbildung 5.5.: Einzige stabile mit MCMC ermittelte Konstellation mit $\sigma_{Pos.} = 0,5$
und $\sigma_{Rot.} = \pi/8$

6. Optimierte MCMC

6.1. Optimierung für Problem 1

Da sich weiter oben bereits gezeigt hat, dass die MCMC-Methode mit der hier verwendeten Funktion $p^*(x) = \frac{1}{s_{Bew.} + 1}$ nicht dazu geeignet ist, in kurzer Zeit eine stabile Konstellation zu finden, modifizieren wir den Algorithmus hier bezüglich der Berechnung von a . Im bisher verwendeten Algorithmus galt $a = \frac{p^*(x^l)}{p^*(x^{l-1})}$.

Da wir eine möglichst stabile Konstellation suchen, wird die Berechnung von a hier derart modifiziert, dass dessen Wert in dem Fall, dass die momentane Probe weniger stabil als die vorhergehende ist, einen kleinen Wert 0,001 annimmt, und im Fall $p^*(x^l) \geq p^*(x^{l-1})$ den Wert 1, um so zu erreichen, dass Proben, die stabiler als ihre Vorgänger sind, bezüglich der Akzeptanz stark bevorzugt werden. Der so modifizierte Algorithmus ist in Algorithm 5 zu sehen.

Algorithm 5 MCMC mit Optimierung 1

```
1: Anzahl Proben  $n_{probe}$  und Toleranz festlegen, Startprobe  $x^1$  simulieren
2: Zu stabilisierenden dynamischen Körper generieren
3: for  $l = 2$  bis  $L$  do
4:   Probe  $x^l$  aus  $\mathcal{N}(x^l | x^{l-1}, \sigma^2 \mathbf{I})$  ziehen
5:    $s_{Start}^l$  speichern und Simulation starten
6:    $s_{Ende}^l$  speichern
7:    $s_{Bew.}^l = \|s_{Ende}^l - s_{Start}^l\|_2$ 
8:   if  $s_{Bew.}^l \geq Toleranz$  then go to 4
9:    $p^*(x^l) = \frac{1}{s_{Bew.}^l + 1}$ 
10:  if  $p^*(x^l) \leq p^*(x^{l-1})$  then
11:    Setze  $a = 0,001$ 
12:  else
13:    Setze  $a = 1$ 
14:  if  $a \geq 1$  then  $x^l$  wird als Probe akzeptiert
15:  else
16:    Ziehe gleichverteilt ein  $u \in [0, 1]$ 
17:    if  $u \leq a$  then  $x^l$  wird als Probe akzeptiert
18:  else
19:    Setze  $x^l = x^{l-1}$ 
```

Tabelle 6.1.: MCMC mit Optimierung 1 in Problem 1 mit 100.000 Proben, $\sigma = 0,5$ und $\sigma_{rot.} = \pi/8$

Versuch	a_{max}	\bar{a}	$s_{Bew.min}$	$\bar{s}_{Bew.}$
1	0,534	0,282	0,417	0,841
2	0,326	0,209	0,903	1,271
3	0,213	0,153	1,362	1,651
4	0,196	0,112	1,592	2,127
5	0,138	0,090	2,181	2,631
6	0,107	0,068	2,667	3,154
7	0,087	0,052	3,057	3,613

Tabelle 6.2.: MCMC mit Optimierung 1 in Problem 1 mit 100.000 Proben, $\sigma = 1$ und $\sigma_{rot.} = \pi/4$

Versuch	a_{max}	\bar{a}	$s_{Bew.min}$	$\bar{s}_{Bew.}$
1	0,386	0,208	0,640	1,024
2	0,308	0,133	0,916	1,502
3	0,194	0,116	1,488	1,920
4	0,147	0,086	1,926	2,399
5	0,098	0,064	2,471	2,837
6	0,070	0,050	3,043	3,328
7	0,075	0,054	3,393	3,775

6.1.1. Versuchsparameter

Die Parameter sind exakt die selben wie schon bei der Durchführung ohne Optimierungen des Algorithmus. Auch hier werden zwei Versuchsreihen mit $\sigma_{Pos.} = 0,5$ und $\sigma_{Rot.} = \pi/8$, beziehungsweise $\sigma_{Pos.} = 1$ und $\sigma_{Rot.} = \pi/4$ durchgeführt.

6.1.2. Ergebnisanalyse

Die Ergebnisse der Versuche sind in den Tabellen 6.1 und 6.2 aufgelistet. Es ist direkt ersichtlich, dass mit der optimierten Funktion a deutlich bessere Ergebnisse erzielt werden. Die Stabilitätswerte der ersten Versuchsreihe sind dabei eindeutig höher, als die der zweiten.

Eine stabile Konstellation wurde jedoch auch hier noch nicht gefunden. Möglicherweise führt eine weitere Optimierung der Funktion a oder eine Erhöhung der Anzahl der gezogenen Proben zu einer letztendlich stabilen Konstellation.

6.1.3. Visualisierung der Versuchsergebnisse

Der zeitliche Verlauf der Stabilitätswerte ist in Abbildung 6.1 (Seite 50) aufgetragen, die Häufigkeit derselben in Abbildung 6.2 (Seite 51).

Es ist zu sehen, dass der Verlauf der Stabilitäten im Vergleich zu den Versuchen weiter oben deutlich glatter ist. Dies ist auf der Verwendung der optimierten Funktion a zurückzuführen, da keine anderweitigen Änderungen am Algorithmus vorgenommen wurden.

In Abbildung 6.2 kann man erkennen, dass die Werte für die Stabilität deutlich höher sind, als noch mit der nicht modifizierten MCMC-Methode.

In Abbildung 6.3 (Seite 52) ist eine Auswahl an Konstellationen zu sehen, die einen verhältnismäßig hohen Stabilitätswert besitzen.

6.2. Optimierung für Problem 2

Zur Verbesserung der Versuchsergebnisse in *Problem 2* mit der MCMC-Methode passen wir an dieser Stelle die Funktion $p^*(x)$ an. Die optimierte Berechnung von $p^*(x)$ findet sich in Algorithm 6 in den Zeilen 8 bis 14.

Für den Fall, dass die gefundene Konstellation eine stabile Brücke bildet, wird die Probe so in jedem Fall akzeptiert. Im Falle einer instabilen Konstellation gibt es zwei Möglichkeiten:

- Die vorhergehende Konstellation war stabil: In diesem Fall wird $p(\mathbf{x}^l) = 0,001$ gesetzt, womit es sehr unwahrscheinlich wird, dass nach einer stabilen Probe eine instabile akzeptiert wird.
- Die vorhergehende Konstellation war instabil: In diesem Fall erhält $p(\mathbf{x}^l)$ einen Zufallswert. Somit werden aufeinander folgende instabile Konstellationen auf zufälliger Basis akzeptiert und verworfen, bis eine stabile Konstellation gefunden wird

6.2.1. Versuchsparameter

Wir verwenden hier die selben Parameter wie in der Anwendung der MCMC-Methode auf *Problem 2* aus Kapitel 5.4.1.

6.2.2. Ergebnisanalyse

Die Ergebnisse der Versuche sind in den Tabellen 6.3 und 6.4 aufgelistet. Es ist zu sehen, dass die Anzahl der gefundenen stabilen Konstellationen deutlich höher ist, als noch bei der Anwendung der MCMC-Methode ohne die modifizierte Berechnung von $p^*(x)$. Jedoch wird auch hier nicht in jedem Fall eine stabile Konstellation gefunden. Außerdem sind viele der Konstellationen miteinander identisch.

Algorithm 6 MCMC mit Optimierung 2

```
1: Anzahl Proben  $n_{probe}$  und Toleranz festlegen, Startprobe  $x^1$  simulieren
2: Zu stabilisierenden dynamischen Körper generieren
3: for  $l = 2$  bis  $L$  do
4:   Probe  $x^l$  aus  $\mathcal{N}(x^l|x^{l-1}, \sigma^2\mathbf{I})$  ziehen
5:    $s_{Start}^l$  speichern und Simulation starten
6:    $s_{Ende}^l$  speichern
7:    $s_{Bew.}^l = \|s_{Ende}^l - s_{Start}^l\|_2$ 
8:   if  $s_{Bew.}^l \geq H$  then
9:     if  $p^*(x^{l-1}) \leq 1$  then
10:      Ziehe  $u$  gleichverteilt aus  $[0, 1]$  und setze  $p^*(x^l) = u$ 
11:     else
12:       Setze  $p^*(x^l) = 0,001$ 
13:     else
14:       Setze  $p^*(x^l) = 1$ 
15:   if  $a \geq 1$  then  $x^l$  wird als Probe akzeptiert
16:   else
17:     Ziehe gleichverteilt ein  $u \in [0, 1]$ 
18:     if  $u \leq a$  then  $x^l$  wird als Probe akzeptiert
19:     else
20:       Setze  $x^l = x^{l-1}$ 
```

Tabelle 6.3.: Optimiertes MCMC in Problem 2 mit $\sigma_{Pos.} = 0,5$ und $\sigma_{Rot.} = \pi/8$

Versuch	Anzahl stabiler Konstellationen	Anzahl voneinander verschiedener stabiler Konstellationen
1	17.518	290
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	7.188	2

Tabelle 6.4.: Optimiertes MCMC in Problem 2 mit $\sigma_{Pos.} = 1$ und $\sigma_{Rot.} = \pi/4$

Versuch	Anzahl stabiler Konstellationen	Anzahl voneinander verschiedener stabiler Konstellationen
1	27.216	97
2	0	0
3	1	1
4	2.452	4
5	0	0
6	0	0
7	0	0

Tabelle 6.5.: Optimiertes MCMC in Problem 2 mit $\sigma_{Pos.} = 0,5$, $\sigma_{Rot.} = \pi/8$ und 1.000.000 gezogenen Proben pro Versuch

Versuch	Anzahl stabiler Konstellationen	Anzahl voneinander verschiedener stabiler Konstellationen
1	4.199	22
2	6.707	5
3	3.982	6
4	641	1
5	5.567	7
6	6.713	3
7	0	0

6.2.3. Visualisierung der Versuchsergebnisse

Der zeitliche Verlauf der Stabilitätswerte ist in Abbildung 6.4 (Seite 53) zu sehen.

6.2.4. Zweite Versuchsreihe

Da die mit dem optimierten MCMC-Algorithmus erzielten Ergebnisse die bisher besten zu *Problem 2* sind, führen wir an dieser Stelle eine weitere Versuchsreihe durch. Die Parameter bleiben die selben, wie zuvor, mit dem Unterschied, dass nun 1.000.000 Proben je Versuch gezogen werden.

6.2.5. Ergebnisse

Die Ergebnisse dieser Versuchsreihe sind in Tabelle 6.5 aufgelistet. Man erkennt, dass in fast allen Versuchen stabile Konstellationen gefunden wurde, dass es jedoch auch vorkommen kann, dass wie im 7. Versuch keine solche Konstellation gefunden wird.

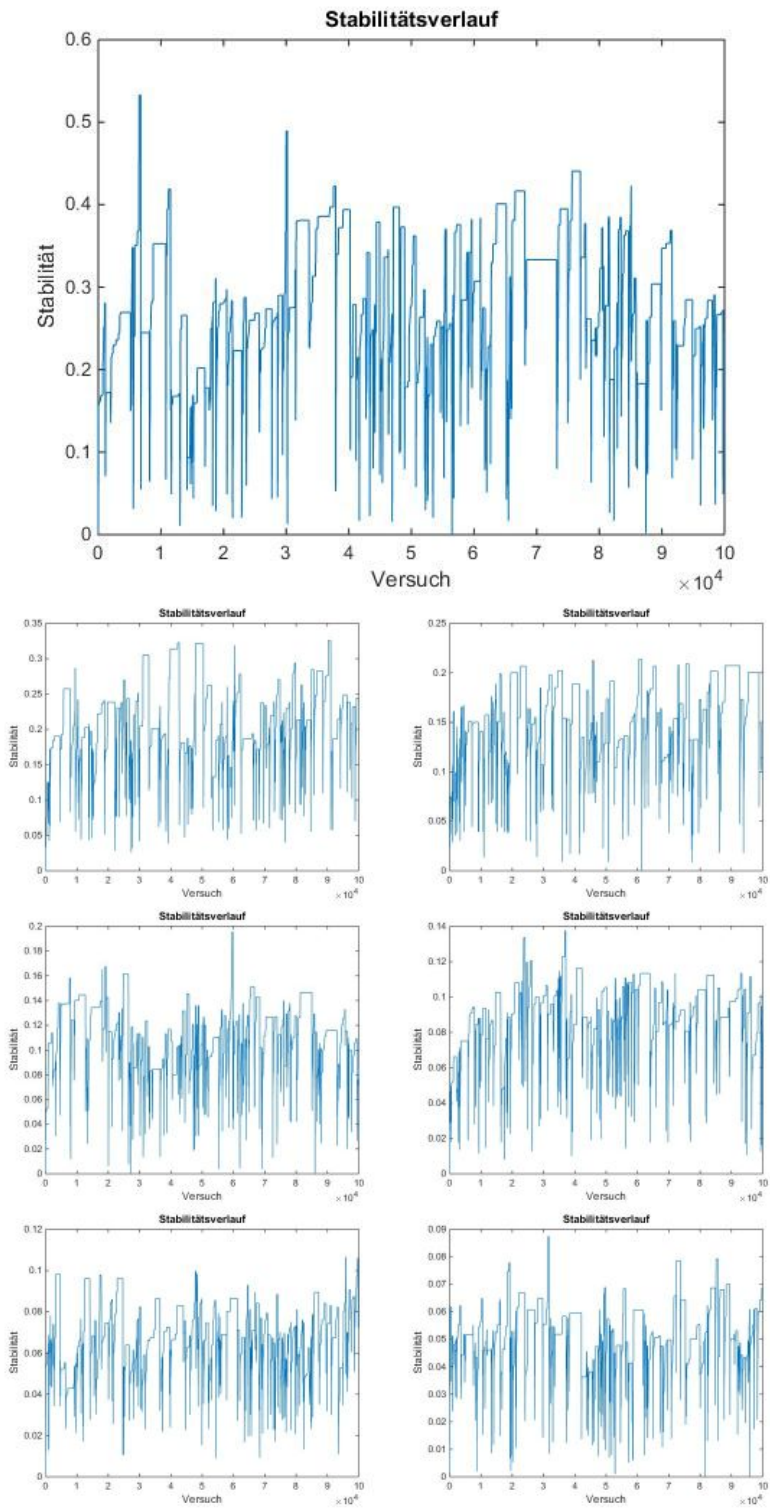


Abbildung 6.1.: Verlauf der Stabilitätswerte in optimierter MCMC-Methode mit $\sigma_{Pos.} = 0,5$ und $\sigma_{Rot.} = \pi/8$

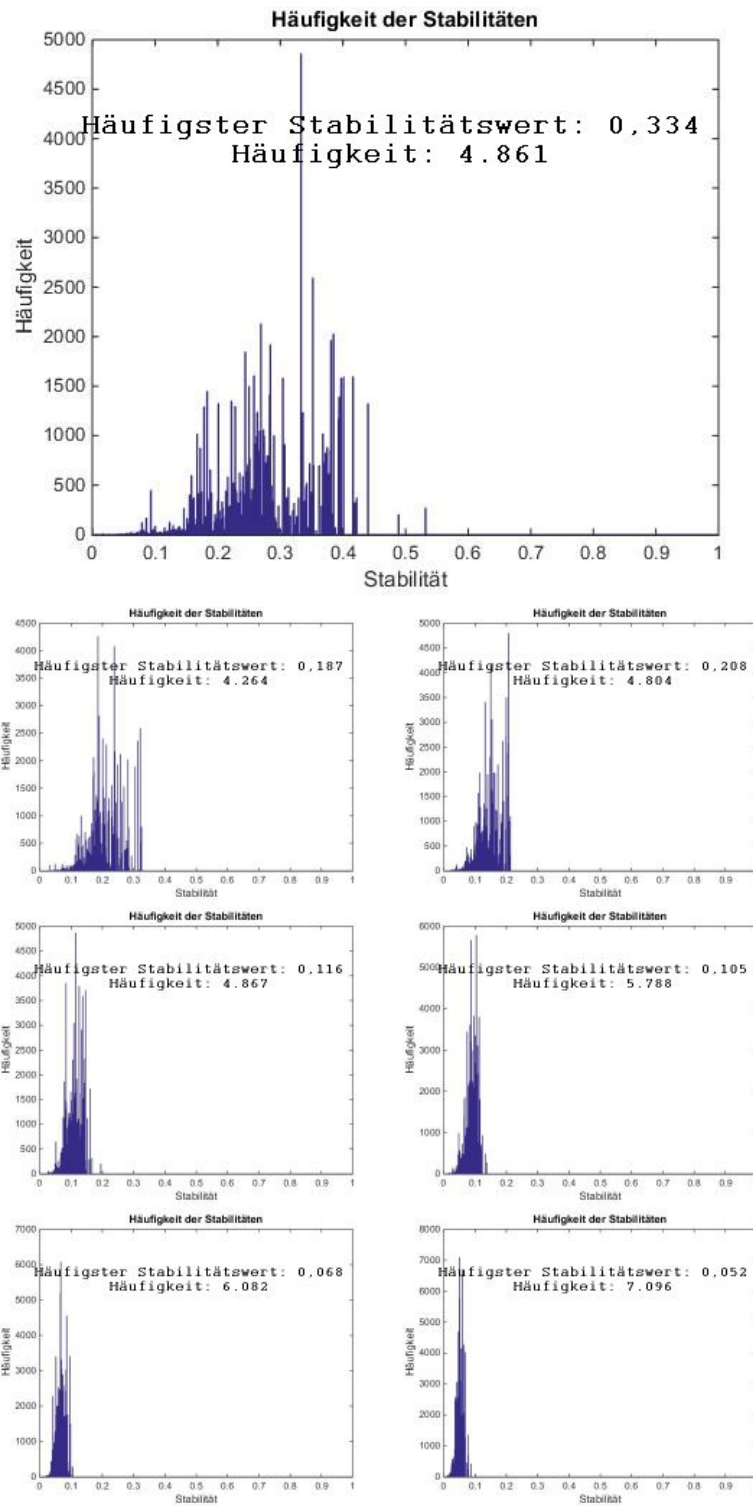


Abbildung 6.2.: Verteilung der Stabilitätswerte in optimierter MCMC-Methode mit $\sigma_{Pos.} = 0,5$ und $\sigma_{Rot.} = \pi/8$

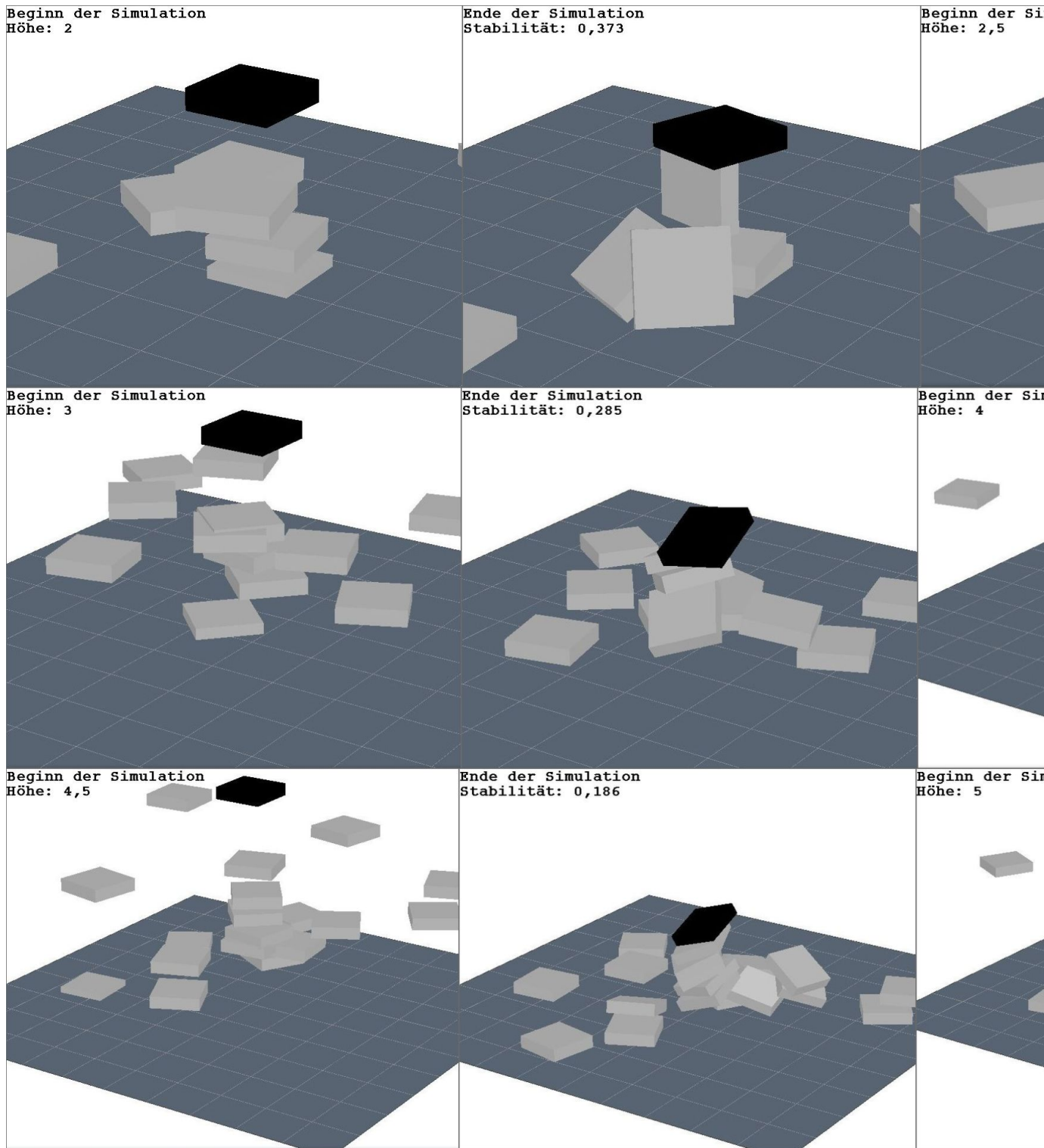


Abbildung 6.3.: Auswahl der Konstellationen mit den höchsten Stabilitätswerten mit $\sigma_{Pos.} = 0,5$ und $\sigma_{Rot.} = \pi/8$ (Stabilitätswerte nicht normiert)

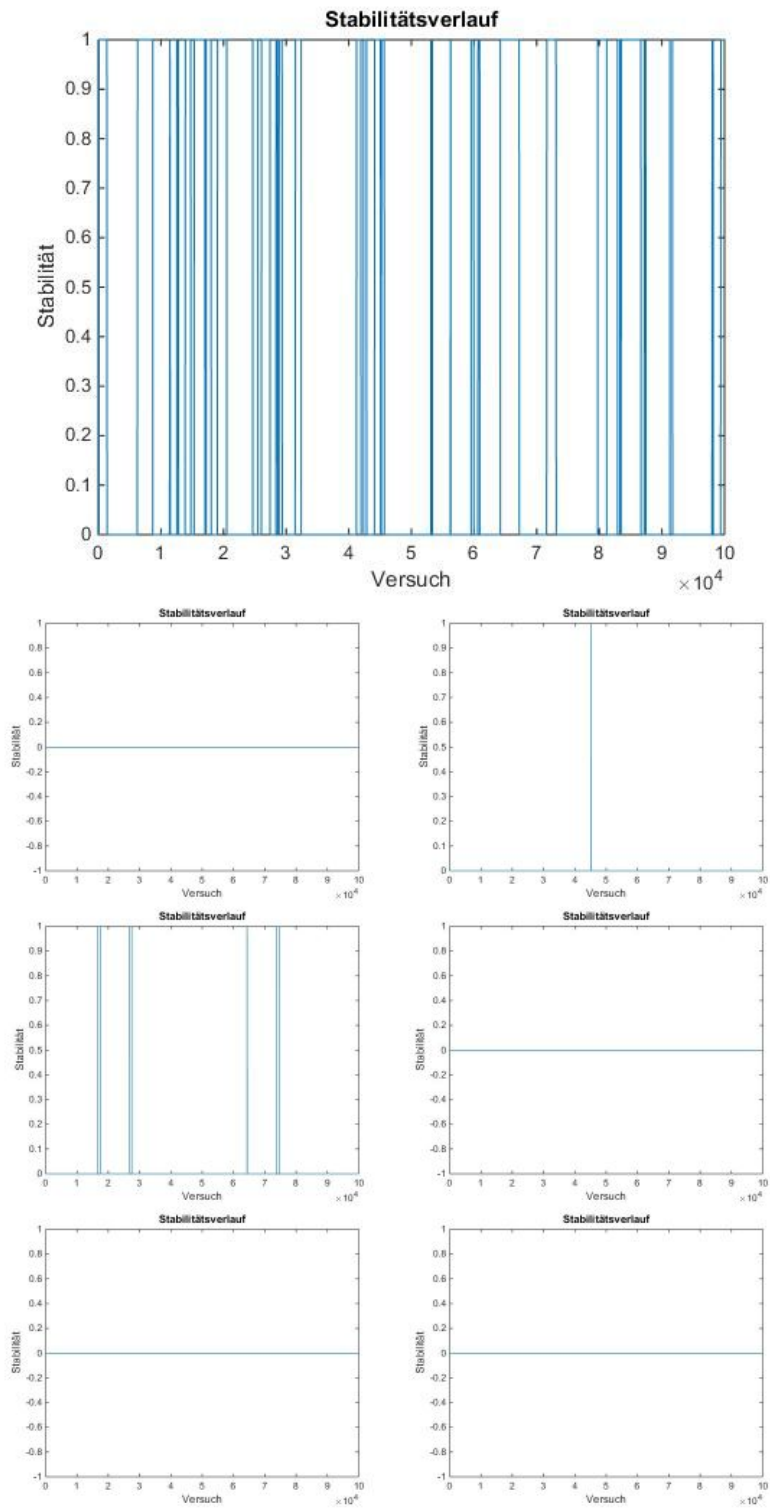


Abbildung 6.4.: Stabilitätsverlauf über der Zeit mit $\sigma_{Pos.} = 1$ und $\sigma_{Rot.} = \pi/4$ (Stabilitätswerte nicht normiert)

7. Vergleich aller Ergebnisse

Abschließend lässt sich sagen, dass das MCMC-Verfahren mit Optimierungen die besten Resultate liefert und einen deutlichen Fortschritt zu Rejection Sampling darstellt.

Allerdings gelingt es auch mit dem modifizierten MCMC-Verfahren nicht zuverlässig, eine stabile Lösung zu *Problem 1* zu finden. Dieser Mangel kann möglicherweise durch weitere Modifikationen oder eine höhere Anzahl gezogener Proben behoben werden.

Da das Ziehen von mehreren hunderttausend Proben jedoch einigen Zeitaufwand erfordert, bietet es sich hier eher an, das Verfahren zu verbessern, zum Beispiel durch das Einführen von physikalischem Vorwissen, das im Algorithmus selbst Anwendung findet.

Literaturverzeichnis

- [1] David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.