

**Großkanonische Monte-Carlo Simulationen
eines Modells harter Stäbchen**
Computational Physics Praktikum WS24/25

Sascha Eckstein, Jannis Hollmann
Version 1

Inhaltsverzeichnis

1	Aufgabe 3.1.1: Entwurf der Datenstruktur	3
1.1	Periodische Randbedingungen	3
1.2	Kollisionstest	3
1.3	Einfügen eines Stäbchens	3
1.4	Entfernen eines Stäbchens	3
1.5	Aktuelle Stäbchenzahlen	3
2	Aufgabe 3.1.2: Detailed balance	4
3	Aufgabe 3.2.1: Teilchenzahlen messen	5
4	Aufgaben 3.2.2 & 3.2.3: Histogramme aufnehmen	6
5	Aufgabe 3.2.4: Resultat: Ordnung vs Dichte	6
6	Aufgabe 3.3: Visualisierung	7

1 Aufgabe 3.1.1: Entwurf der Datenstruktur

Die in unserem Konzept verfolgte Idee zur Datenspeicherung ist die Folgende: Zunächst wird ein Struct `Rod` definiert, das je die Koordinaten x, y sowie die Ausrichtung $s \in \{-1, 1\}$ beinhaltet. Diese werden dann in einem eindimensionalen Vektor `rods` gespeichert. Um zu überprüfen, ob ein neu hinzugefügtes Stäbchen platziert werden darf, wird zusätzlich zu dem 1d-Vektor `rods` auch ein 2d-Array von Integeren der Größe $M \times M$ namens `grid` gespeichert. Dessen zwei Dimensionen sind die zwei räumlichen Dimensionen des simulierten Gitters. Die gespeicherten Integeren sind -1 , sofern sich an dem jeweiligen Punkt kein Stäbchen befindet. Falls sich die Ausbreitung eines Stäbchens über einen Punkt ausbreitet (also nicht nur seine Koordinaten dort liegen), wird im jeweiligen Kästchen der Index des Stäbchens aus dem Vektor `rods` gespeichert. Die Integrität und Übereinstimmung der beiden Datenstrukturen wird sichergestellt, indem das Einfügen und Entfernen von Stäbchen nur von spezialisierten Methoden ausgeführt wird und nicht durch direkte Manipulation erfolgt. Diese Methoden verwalten auch die drei Integeren `horizontalRods`, `verticalRods` und `totalRods`. Diese werden zur Reduktion des Berechnungsaufwands verwendet.

1.1 Periodische Randbedingungen

Für die Behandlung der periodischen Randbedingungen wird durch die Anwendung des Modulo

$$(x + M) \bmod M, \quad (1)$$

dass alle Koordinaten $x \geq M$ auf die gegenüberliegende Seite gewrappert werden. Um sich gegen potentiell negative Koordinaten abzusichern, wird zuerst noch M addiert.

1.2 Kollisionstest

Für den Kollisionstest wird in der Funktion `canInsertRod(x, y, s)` eine `for`-Loop durchlaufen, die vom Punkt des geplanten Einsetzens (x, y) aus entsprechend der Ausrichtung s startet und bis zur Länge L der Stäbchen jeden darauffolgenden Gitterpunkt von `grid` auf Besetztheit überprüft. Sobald einer der Punkte -1 enthält (besetzt ist), gibt die Funktion sofort `false` zurück, falls jedoch die Schleife durchläuft, wird stattdessen `true` zurückgegeben.

1.3 Einfügen eines Stäbchens

Falls es möglich ist ein Stäbchen einzusetzen, wird die Funktion `insertRod(x, y, s)` aufgerufen. Diese fügt zunächst das Stäbchen zum Vektor `rods` hinzu und durchläuft dann nach demselben Prinzip wie beim Kollisionstest das `grid`, um dann den Index des neuen Stäbchens in `rods` in die besetzten Gitterpunkte von `grid` zu speichern. Zudem wird die entsprechende Variable `horizontalRods` oder `verticalRods` sowie `totalRods` inkrementiert.

1.4 Entfernen eines Stäbchens

In `removeRod(index)` wird zunächst das Stäbchen `r` anhand seines Index aus dem Vektor `rods` gelesen. Dann wird das Gitter `grid` an den Positionen, an denen sich das Stäbchen befand, durchlaufen und jeder Wert auf -1 gesetzt. Ebenso wird entsprechend `horizontalRods` bzw. `verticalRods` und `totalRods` dekrementiert. Zum Schluss wird das Stäbchen noch aus dem Vektor `rods` gelöscht.

1.5 Aktuelle Stäbchenzahlen

Für die aktuellen Stäbchenzahlen wurde ursprünglich über den Vektor `rods` iteriert und für jedes Stäbchen überprüft, welche Ausrichtung es hat, um am Ende die Summe zurückgeben zu können. Dies wurde aus Performancegründen durch drei Zählervariablen ersetzt, die nur minimal zusätzlichen Speicher verursachen, dafür jedoch deutlich weniger rechenintensiv sind. Man könnte auch nur zwei (für horizontale und vertikale Stäbchenzahl) verwenden und für die Gesamtzahl diese beiden Zahlen addieren, aber da der zusätzliche Speicherverbrauch lieber in Kauf genommen wird als die Addition, wurde die dritte Variable für die Gesamtzahl eingeführt.

2 Aufgabe 3.1.2: Detailed balance

Es soll gezeigt werden, ob die Akzeptanzwahrscheinlichkeiten für das Einfügen und Entfernen von Stäbchen

$$\alpha_{i \rightarrow j}^{\text{ins}} = \min \left(1, \frac{2M^2}{N+1} e^{\beta\mu} e^{-\beta\Delta U} \right) \quad (2)$$

$$\alpha_{j \rightarrow i}^{\text{del}} = \min \left(1, \frac{N+1}{2M^2} e^{-\beta\mu} e^{\beta\Delta U} \right) \quad (3)$$

die *detailed balance condition*

$$p'(\omega'_i) q_{i \rightarrow j} = p'(\omega'_j) q_{j \rightarrow i} \quad (4)$$

erfüllen. Hierbei ist $q_{i \rightarrow j}$ die Übergangsrate von der Konfiguration i zur Konfiguration j und $p'(\omega'_i)$ die Wahrscheinlichkeit für die Konfiguration i .

Zunächst werden die Übergangsraten zerlegt in eine Vorschlagswahrscheinlichkeit π und eine Akzeptanzwahrscheinlichkeit α :

$$q_{i \rightarrow j} = \pi_{i \rightarrow j} \cdot \alpha_{i \rightarrow j}. \quad (5)$$

Hiermit lässt sich die *detailed balance condition* (4) umformen zu

$$\frac{p'(\omega'_i)}{p'(\omega'_j)} = \frac{\pi_{j \rightarrow i} \cdot \alpha_{j \rightarrow i}}{\pi_{i \rightarrow j} \cdot \alpha_{i \rightarrow j}} \quad (6)$$

$$\Leftrightarrow x \equiv \frac{\pi_{i \rightarrow j}}{\pi_{j \rightarrow i}} \frac{p'(\omega'_i)}{p'(\omega'_j)} = \frac{\alpha_{j \rightarrow i}}{\alpha_{i \rightarrow j}}. \quad (7)$$

Wählen wir nun

$$\alpha_{i \rightarrow j} = \min(1, x) \quad (8)$$

$$\alpha_{j \rightarrow i} = \min \left(1, \frac{1}{x} \right). \quad (9)$$

Dies ist erlaubt, da

$$x = \frac{\min(1, x)}{\min(1, \frac{1}{x})} = \begin{cases} x & x < 1 \\ \frac{1}{x} & x \geq 1. \end{cases} \quad (10)$$

Zurück zur anderen Seite von (7): Wir wissen, dass für die Vorschlagsraten

$$\pi_{i \rightarrow j}^{\text{ins}} = \frac{1}{4M^2} \quad (11)$$

$$\pi_{j \rightarrow i}^{\text{del}} = \frac{1}{2} \quad (12)$$

gilt, unter der Annahme, dass i die Konfiguration mit N Stäbchen und j die Konfiguration mit $N+1$ Stäbchen ist. Im großkanonischen Ensemble gilt $p'(\omega'_i) = \exp(\beta\mu N)/N!$ unter der Annahme, dass die Wechselwirkungsenergie 0 ist. Dies ist hier der Fall, da wir vor dem Einsetzen eines Stäbchens Kollisionen ausschließen. Wir erhalten also

$$\begin{aligned} x &= \frac{\pi_{i \rightarrow j}^{\text{ins}} p'(\omega'_i)}{\pi_{j \rightarrow i}^{\text{del}} p'(\omega'_j)} \\ &= \frac{\frac{1}{4M^2} \frac{1}{(N+1)!} \exp(\beta\mu(N+1))}{\frac{1}{2} \frac{1}{N!} \exp(\beta\mu N)} \\ &= 2M^2 \frac{\exp(\beta\mu N)}{N+1} \end{aligned} \quad (13)$$

Durch Einsetzen in (8) und (9) erhält man schließlich

$$\alpha_{i \rightarrow j}^{\text{ins}} = \min \left(1, \frac{2M^2}{N+1} e^{\beta\mu} \right) \quad (14)$$

$$\alpha_{j \rightarrow i}^{\text{del}} = \min \left(1, \frac{N+1}{2M^2} e^{-\beta\mu} \right). \quad (15)$$

3 Aufgabe 3.2.1: Teilchenzahlen messen

In dieser Aufgabe soll ermittelt werden, nach wie vielen Schritten das System als thermalisiert betrachtet werden kann. Hierzu wurde der Algorithmus für $1.5 \cdot 10^8$ Schritte laufen gelassen und alle 10^4 Schritte die Anzahl der Stäbchen $\{N, N_+, N_-\}$ gespeichert, wobei N_+ die horizontalen und N_- die vertikalen Stäbchen sind. Dies wurde für die drei vorgegebenen Aktivitäten $e^{\beta\mu} \equiv z \in \{0.56, 0.84, 1.1\}$ wiederholt. Die Ergebnisse sind in Abb. 1 dargestellt.

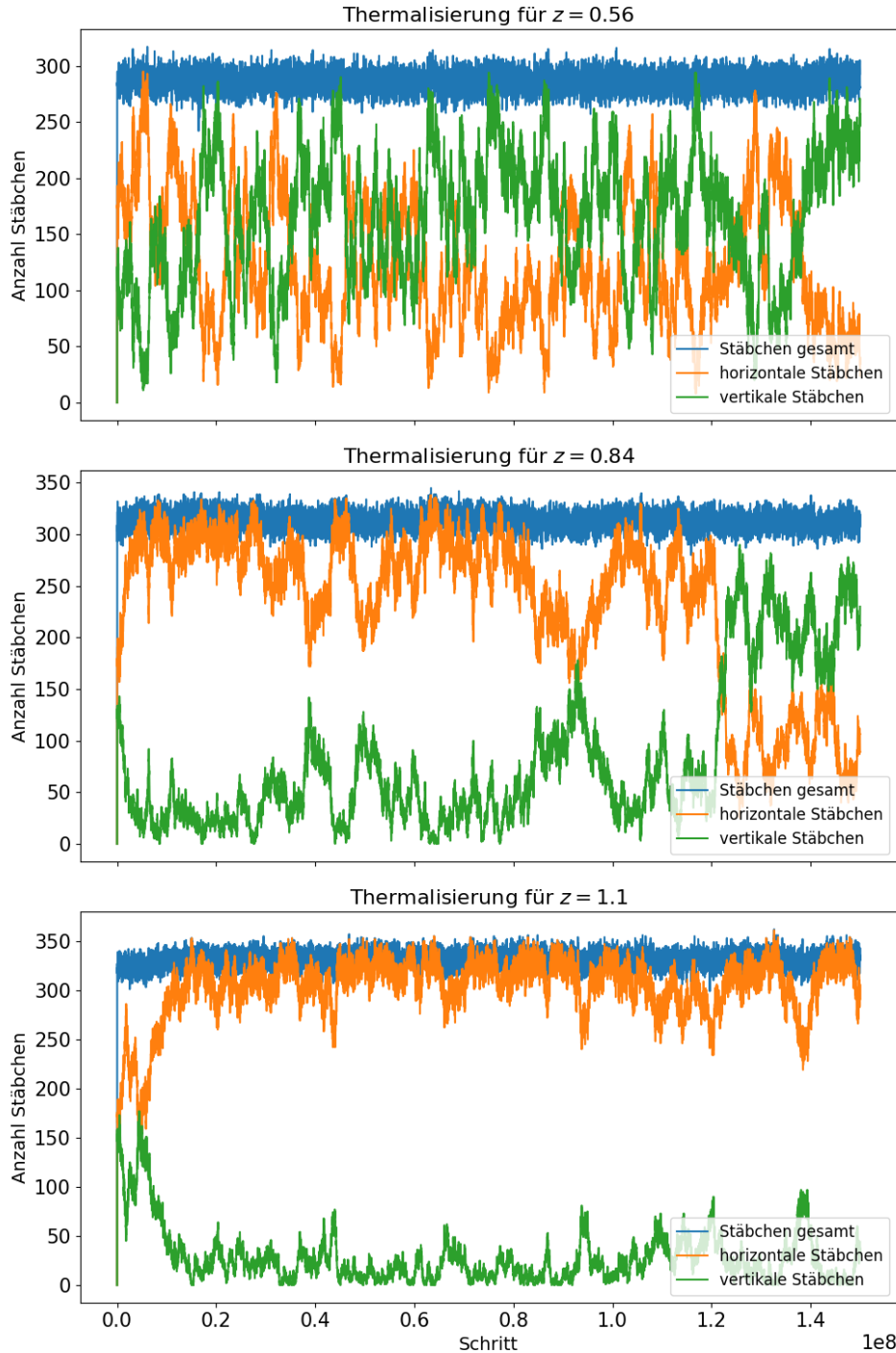


Abbildung 1: Verlauf der Stäbchenzahlen

An allen drei Plots sieht man gut, dass die Gesamtstäbchenzahl N relativ schnell auf einen Wert ansteigt, der zwar statistischen Oszillationen unterliegt, deren Mittelwert aber doch sehr stabil erscheint. Ab diesem Punkt kann angenommen werden, dass das System thermalisiert ist. Insbesondere am mittleren Plot (für $z = 0.84$) sieht man, dass die Stäbchenzahlen N_+ und N_- während des allerersten Anstiegs im Verhältnis 1 : 1 stehen. Sobald die Gesamtstäbchenzahl ihren stabilen Wert erreicht, beginnt eine der Stäbchenzahlen (hier N_-) an, zu

fallen, während die andere weiter ansteigt. Das bedeutet, dass das System nun phasenseparierend ist. Man sieht, dass diese Phasenseparation bei höherer Aktivität stabiler ist, der Plot der niedrigsten Aktivität zeigt starke Schwankungen darin, welche Ausrichtung die Mehrheit ist. Auch bei der höchsten Aktivität wechseln sich die Orientierungen ab, dies wurde im Plot jedoch abgeschnitten, um den Anfangsbereich besser sehen zu können.

4 Aufgaben 3.2.2 & 3.2.3: Histogramme aufnehmen

In diesen beiden Aufgaben sollen Histogramme der verschiedenen Observablen bei verschiedenen Aktivitäten z erstellt werden. Hierfür wird zunächst thermalisiert und dann $4 \cdot 10^9$ Schritte lang simuliert und dabei Daten aufgezeichnet. Diese große Dauer der Aufzeichnung ist wichtig, da die Daten sonst keine ausreichende Statistik aufweisen - bei einem Versuch zeigte sich, dass für $z = 0.84$ und $z = 1.1$ nur $S > 0.5$ auftrat und somit nie die vertikalen Stäbchen dominierten. Bei längerer Aufzeichnung hingegen konnte die dominierende Ausrichtung auch wechseln.

Man kann den Histogrammen für die horizontalen und vertikalen Stäbchenzahlen N_+ und N_- ansehen, dass bei größerer Aktivität die Verteilung dazu tendiert, dass entweder vertikale oder horizontale Stäbchen dominieren. Dies sieht man an der U-Form der Histogramme. Bei niedrigem z hingegen ist es eher eine Glockenform, was darauf hindeutet, dass häufige Wechsel der dominierenden Ausrichtung stattfinden.

Ebenfalls nützlich zur Charakterisierung des Systems ist der Ordnungsparameter $S = \frac{N_+ - N_-}{N}$. S liegt im Intervall $[-1, 1]$, wobei $S = 1$ bedeutet, dass alle Stäbchen horizontal ausgerichtet sind und $S = -1$ entsprechend, dass alle Stäbchen vertikal ausgerichtet sind. Für $S = 0$ sind beide Ausrichtungen gleich vorhanden. Dies lässt sich sehr gut an den Histogrammen ablesen: In der Aufnahme mit $z = 1.1$ gab es bspw. einen Überschuss an vertikalen Stäbchen (der Peak der vertikalen Stäbchen ist nahe bei 300), deshalb ist der Peak von S bei -1 . Bei $z = 0.56$ sind die Maxima der beiden Ausrichtungen etwa in der Mitte, und daher ist das Maximum von S bei 0.

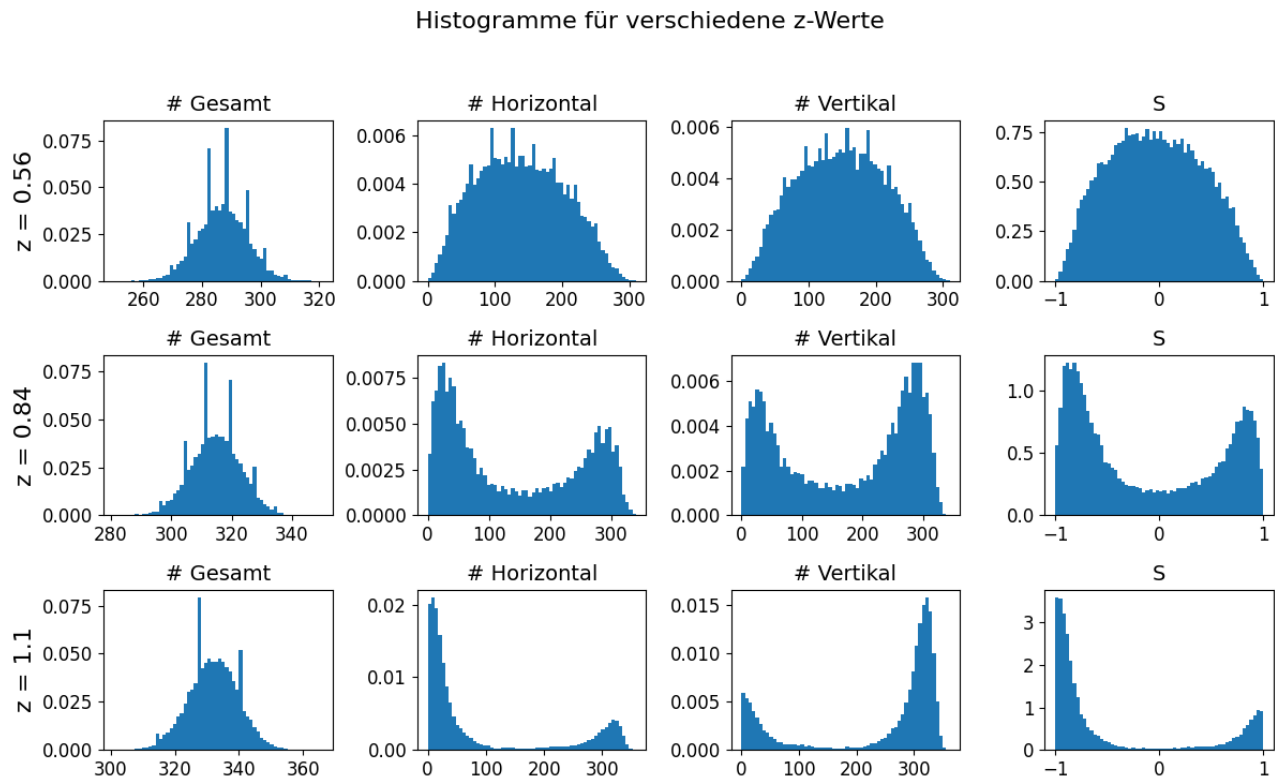
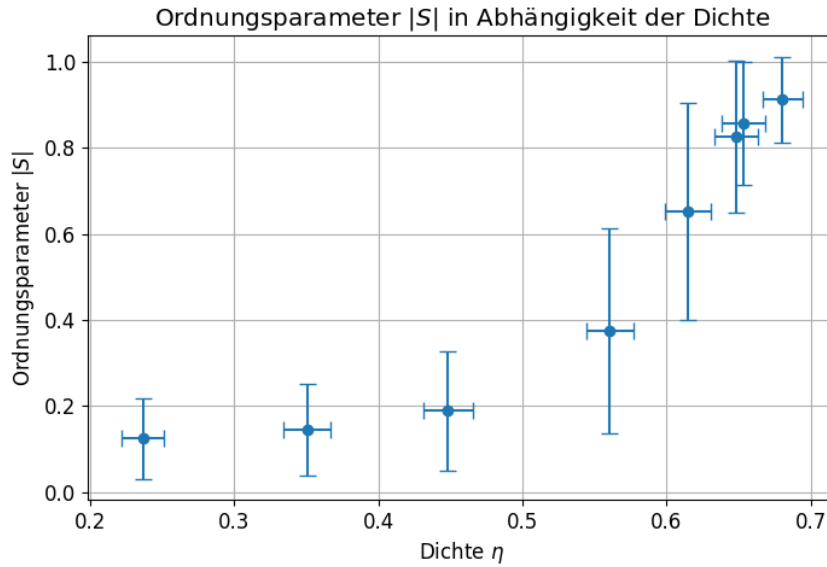


Abbildung 2: Histogramme für verschiedene Aktivitäten z

5 Aufgabe 3.2.4: Resultat: Ordnung vs Dichte

In dieser Aufgabe wird der Zusammenhang zwischen Ordnung und Dichte untersucht. Hierzu werden für zusätzliche Aktivitäten z Daten aufgenommen, um mehr verschiedene Werte für η zu erhalten. Für jede der Aufnahmen wird dann der Mittelwert und die Standardabweichung von η und $|S|$ berechnet und in Abb. 3 dargestellt. Man erkennt, dass eine größere Dichte im Zusammenhang mit einem größeren $|S|$ steht. Das ergibt anschaulich Sinn, da bei zufällig ausgerichteten Stäbchen das Risiko für Lücken größer ist als bei parallelen Stäbchen.

Abbildung 3: Ordnungsparameter $|S|$ in Abhängigkeit der Packungsdichte η

Im thermodynamischen Limes $M \rightarrow \infty$ würde der Phasenübergang schärfer werden. Grund dafür ist, dass bspw. Fluktuationen im kleinen Gitter eine Auswirkung haben, deren Effekt bei größeren Gittern jedoch nachlässt. Die Standardabweichung der Messwerte x_i ist definiert als

$$\sigma = \sqrt{\frac{\langle x_i^2 \rangle - \langle x_i \rangle^2}{N}} = \sqrt{\frac{\langle x_i^2 \rangle - \langle x_i \rangle^2}{N_{\text{MC}}/m}}, \quad (16)$$

wobei N die Anzahl der Messpunkte und entsprechend $N_{\text{MC}} = 4 \cdot 10^9$ die Gesamtzahl der durchgeführten Simulationsschritte und $m = 100\,000$ die Anzahl ausgeführter Schritte zwischen Messungen der Observablen ist. Wenn man nun die Simulation mehrfach laufen lässt und jedes Mal einen Mittelwert eines Messwerts berechnet, werden diese Mittelwerte die Standardabweichung

$$\sigma_M = \frac{\sigma}{\sqrt{\frac{N_{\text{MC}}-1}{m}}} \quad (17)$$

haben. Beide Fehler gehen gegen null, wenn man die Anzahl der Messungen gegen unendlich gehen lässt: Da sich die Anzahl Messungen im Nenner befinden, sinken die Standardabweichungen.

6 Aufgabe 3.3: Visualisierung

In dieser Aufgabe soll ein bereitgestelltes Python-Skript genutzt werden, das eine bestimmte Stäbchenkonfiguration einliest und diese plottet. Es erfordert zwei Dateien "Senkrechte.dat" sowie "Waagerechte.dat", die jeweils die x- und y-Positionen der Stäbchen enthalten. Um diese Dateien zu erhalten, wurde das Simulationsprogramm so modifiziert, dass es die passenden Dateien erzeugt. Zwei Situationen werden hier beispielhaft gezeigt: Der Koexistenzpunkt in Abb. 4 und der einphasige Bereich in Abb. 5.

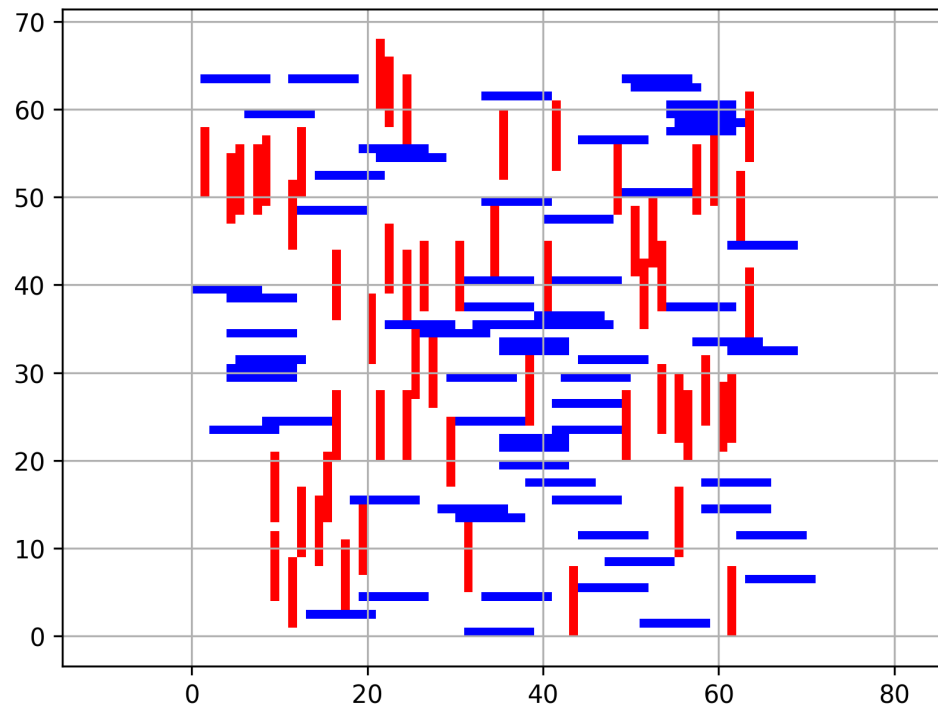


Abbildung 4: Darstellung der Stäbchen am Koexistenzpunkt

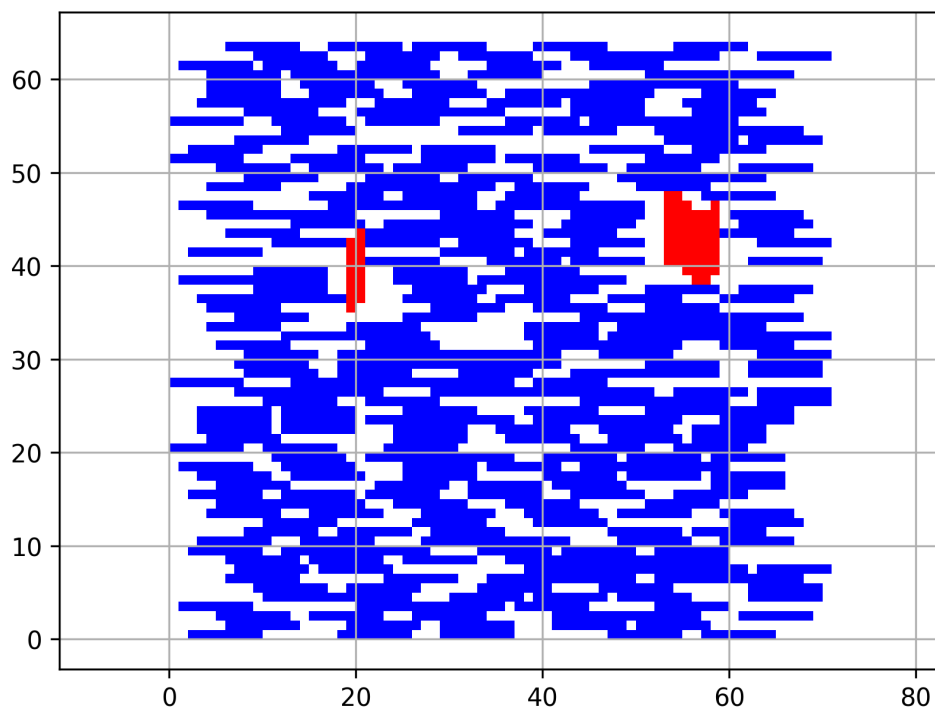


Abbildung 5: Darstellung der Stäbchen in einphasigen Bereich