6. Übung

Theoretische Physik
Universität des Saarlandes
Prof. Dr. Heiko Rieger und
Dr. Adam Wysocki

(Abgabe: bis zum *14. Juni 2017, 16:00 Uhr*. Quellcode, Filme und Bilder bitte in "/home/comphys/comphys_ss17_Abgabe/" im Cip Pool ablegen. Der schriftliche Teil kann entweder als Pdf beigelegt oder im Postfach von Prof. Rieger abgegeben werden.
Unter "/home/comphys/comphys_ss17/exercises_supplemental/" finden sie die jeweilig Dateien, die für die Bearbeitung hilfreich sind.
In der Übung am 14. Juni werden die Lösungen zu Blatt 5 und 6 besprochen.

Präsenzübungen/Einführung:

Das Programm *mdbasic_0* simuliert ein mikrokanonisches Ensemble von Soft-Core-Teilchen in zwei Dimensionen mit periodischen Randbedingungen. Die Simulationsparameter werden zu Laufzeitbeginn aus der Parameterdatei *mdbasic_0.in* eingelesen. Die Einheiten sind entdimensionalisiert, d.h $m = \sigma = \epsilon = 1$:

Lennard-Jones-Potential: $$u(r_{ij}) = 4\epsilon\left(\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right) + \epsilon \qquad r_{ij} \leq r_c = 2^{1/6}\sigma$$

(a) Öffnen Sie die Quellcodedatei **mdbasic_0.c** und versuchen Sie die Funktionsweise grob zu überblicken. Zu dem Programm ist ebenfalls eine Parameterdatei **mdbasic_0.in** vorhanden.

(b) Kompilieren Sie das Programm **./compile.sh 0**. Führen Sie das Programm anschließend aus. Die späteren Programmteile können Sie mit *./compile.sh 1* und *./compile.sh 2* kompilieren.

(c) Es wurde eine Bildausgabe implementiert, die Schnappschüsse der Konfigurationen zu regelmäßigen Zeiten (alle *stepSnap* Schritte) in dem Ordner *moviedata* speichert. Um einen Film aus den erzeugten Bildern zu gennerieren, können Sie **./encode.sh** ausführen. Die erzeugte Videodatei *output.avi* lässt sich beispielsweise mit **mplayer output.avi** wiedergeben. Bevor neue Bilder durch das Programm generiert werden, sollten alle Dateien im Ordner *moviedata* gelöscht werden, falls daraus ein Film erzeugt werden soll. Ansonsten werden die allen Bilder am Ende angehangen, sofern eine führere Simulation mehr Bilder erzeugt hatte.

(d) Die Variable *rCut* beschreibt die Länge, ab der das Lennard-Jones Potenial abgeschnitten wird. Da das Programm zur Zeit Soft-Core-Teilchen simuliert, wurde *rCut* genau so gewählt, dass es dem Mininum des Potentials entspricht und so der attraktive Teil weggeschnitten wurde (siehe oben). Wählen Sie ein größeres *rCut*. So können Sie Aggregation erzeugen. Probieren Sie es aus!

(e) Ändern Sie das Programm so ab, dass *rCut* aus der Parameterdatei eingelesen wird.

Aufgaben:

(Bitte für jede Aufgabe jeweils die Quelltextdatei und die Inputdatei abgeben und ggf. Diagramme.)

1. [*4 Punkte*] **Andersen-Thermostat (**Bitte die Dateien *mdbasic_1.c* und *mdbasic_1.in* nennen**)**

In Experimenten wird in der Regel die Temperatur festgehalten ((N,V,T)-Ensemble) und nicht die innere Energie. Um auch das kanonische Ensemble mittels MD-Simulationen zu untersuchen, bedient man sich sog. Thermostate, die das System auf die gewünschte Temperatur bringen und dort halten. In dieser Aufgabe soll das wohl einfachste Thermostat, das Andersen-Thermostat, implementiert und getestet werden. Bei dieser Methode werden im Laufe der Simulation zufällig Teilchen ausgewählt und deren Geschwindigkeitskomponenten gemäß der Wahrscheinlichkeitverteilung, die bei der gewünschten Zieltemperatur vorliegen würde, neu gesetzt.

1 (a) Es ist plausibel anzunehmen, dass die zeitliche Änderung der Temperatur $T'(t)$ zu einer Zeit t proportional zu der Differenz zwischen der momentanen Temperatur $T(t)$ und der Zieltemperatur T_z ist. Für diesen Fall beschreibt die folgende Differentialgleichung den Verlauf der Temperatur

$$\frac{\mathrm{d}T(t)}{\mathrm{d}t} = -c\,(T(t) - T_z)\ .$$

Lösen Sie diese DGL für den Anfangswert $T(0) = T_0$.

0.5 (b) Fügen Sie zwei Parameter (*stepTherm* und *targetTemp*) in das Programm ein und verwenden Sie auch hier die Parameterdatei, anstatt dem Parameter direkt im Programm einen Wert zuzuweisen. *stepTherm* sollen bestimmen, nach wie vielen Schritten eine virtuelle Teilchenkollision vorgenommen wird und *targetTemp* gibt die Zieltemperatur an.

0.5 (c) Schreiben Sie die Funktion *void Thermalize ()*, die ein zufälliges Teilchen des Ensembles auswählt und setzen Sie dessen Geschwindigkeitskomponenten entsprechend einer Normalverteilung mit einer Halbwertsbreite, die der Quadratwurzel der Zieltemperatur entspricht, neu und rufen Sie die Funktion an einer geeigneten Stelle aus. Die Zufallsgeneratoren: *RandInt(int)* und *RandGauss(double)* sind bereits implementiert können ohne Weiteres verwendet werden.

2 (d) Überprüfen Sie nun die Lösung der DGL aus dem ersten Teil der Aufgabe, indem Sie Simulationen (z.b. mit targetTemp = 1) für mindestens drei verschiedene *stepTherm* Werte durchführen und den Temperaturverlauf in geeigneter halblogarithmischer Auftragung darstellen. Bestimmen Sie jeweils c graphisch aus ihren Daten.

Hinweis: In zwei Dimensionen entspricht die Temperatur in den verwendeten Einheiten genau der kinetischen Energie.

2. [*6 Punkte*] **Paarkorrelationsfunktion/Radial distribution function (Rdf) (***mdbasic_2.c* und *mdbasic_2.in***)**
Ein wichtiges Hilfsmittel um Informationen über die Phase des Systems zu bekommen ist Paarkorrelationfunktion:

$$g(r_n) = \frac{Ah_n}{\pi N_a^2 r_n \Delta r},$$

wobei:

A: Systemfläche (Volumen in $2d$)
h_n: normiertes Histogramm (siehe Aufgabenteil a))
N_a: Teilchenanzahl

$$\Delta r = \frac{\text{maxRangeRdf}}{\text{sizeHist}}$$
$$r_n = (n - \frac{1}{2})\Delta r$$

Die Rdf beschreibt die Dichteverteilung in Abhängigkeit vom Abstand eines bestimmten Teilchens.

2 (a) Berechnen Sie mit Hilfe eines Histogramms h_n die Paarkorrelationsfunktion im kanonischen Ensemble. Das Histogramm soll die Anzahl der Teilchenpaare mit einem Abstand r im Intervall $(n-1)\Delta r \leq r < n\Delta r$ speichern. Beachten Sie, dass Sie dem System eine ausreichende Zeit zum äquilibrieren lassen, bevor Sie das Histogramm füllen und mitteln Sie dann über mehrere Zeiten, um eine aussagekräfige Rdf zu erhalten.

2 (b) Geben Sie die Daten in eine Datei aus. Zum Erstellen einer Ausgabedatei können Sie genauso vorgehen, wie es für die Datei *output.dat* bereits im Quellcode geschehen ist. Variieren Sie die Simulationsparameter und plotten Sie anschließend die Paarkorrelationsfunktion für die drei Aggregatzustände und zeigen Sie entsprechend, wenn keine Korrelation besteht, nur im Nahbereich und im Nah- und Fernbereich.
Hinweis: Für die feste Phase müssen die Teilchen aggregieren (s. Einführungsaufgabe). Begünstigt wird dieser Zustand durch hohe Dichten und niedrige Temperaturen.

2 (c) Für die feste Phasen ordnen sich die Teilchen entsprechend der dichtesten Kugelpackung in zwei Dimensionen. Angenommen der Abstand zum nächsten Nachbarn innerhalb dieser Kugelpackung werde als a bezeichnet: Berechnen Sie die Abstände der zweit-, dritt-, und viertnächsten Nachbarn in dieser Kugelpackung. Diese Abstände sollten auch in festen (aggregierten) Phasen anhand der Rdf gut zu erkennen sein. Überprüfen, ob die Maxima der Funktion mit Ihrer Rechnung vereinbar ist. Fertigen Sie dazu ein Diagramm an indem Ihre Rdf und Ihre Ergebisse für die Maximas eingezeichnet sind.

Liste hilfreicher Makros und Variablen, die im Quellcode verwendet werden:

nMol: Anzahl der Teilchen
DO_MOL: *for(n = 0; n < nMol; n ++)*
mol[i].r: Ortsvektor des i-ten Teilchens
mol[i].rv: Geschwindigkeisvektor des i-ten Teilchens (ra statt rv für Beschleunigung entsprechend)
mol[i].r.x: x-Komponente des Ortes des i-ten Teilchens
region.x: Systemlänge in x Richtung (y statt x entsprechend)