

Theoretische Modellierung und Simulation

Übungsblatt Nr. 5, 01.06.2020

Die Übungsblätter können heruntergeladen werden von

<http://www.uni-ulm.de/theochem/>

Abgabe der Lösungen des Übungsblattes bis zum 10.06.2020 auf der Moodle-Plattform

Aufgabe 10: Leapfrog Algorithmus

Beim Leapfrog Algorithmus (“Froschhüpfen”) zur Berechnung von Trajektorien werden die Geschwindigkeiten \vec{v} und die Orte \vec{r} sukzessiv berechnet durch

$$\vec{v}\left(t + \frac{\Delta t}{2}\right) = \vec{v}\left(t - \frac{\Delta t}{2}\right) - \left(\frac{\nabla V}{m}\right)_t \Delta t + \dots \quad (1)$$

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \vec{v}\left(t + \frac{\Delta t}{2}\right) \Delta t + \dots \quad (2)$$

Leiten Sie den Leapfrog Algorithmus her.

Hinweis: Benutzen Sie Taylorentwicklungen der Geschwindigkeit \vec{v} für $(t + \frac{\Delta t}{2})$ und $(t - \frac{\Delta t}{2})$ um den Zeitpunkt t und des Ortes \vec{r} für $(t + \Delta t)$ und (t) um den Zeitpunkt $(t + \frac{\Delta t}{2})$.

Laden Sie eine Skizze Ihrer Lösung auf der Moodle-Plattform hoch.

Aufgabe 11: Molekulardynamiksimulationen

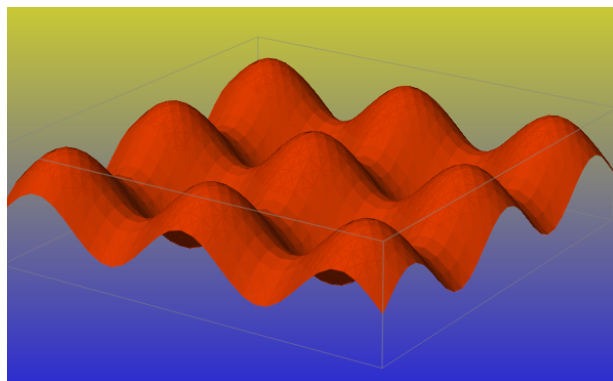
Molekulardynamik (MD) bezeichnet Computersimulationen, bei denen bei gegebenen Wechselwirkungspotential zwischen Atomen und Molekülen die zeitliche Entwicklung des System durch das Lösen von Bewegungsgleichungen bestimmt wird.

Zur Übung sollen Sie die Diffusion eines Teilchens auf einer Oberfläche simulieren, wobei das Potential gegeben ist durch

$$V(x, y) = \frac{1}{4} V_0 \left(2 + \cos\left(\frac{2\pi}{a} x\right) + \cos\left(\frac{2\pi}{a} y\right) \right) \quad (3)$$

mit $a = 3 \text{ \AA}$ und $V_0 = 0.1 \text{ eV}$.

Hier sehen Sie eine Ansicht des Potentials:



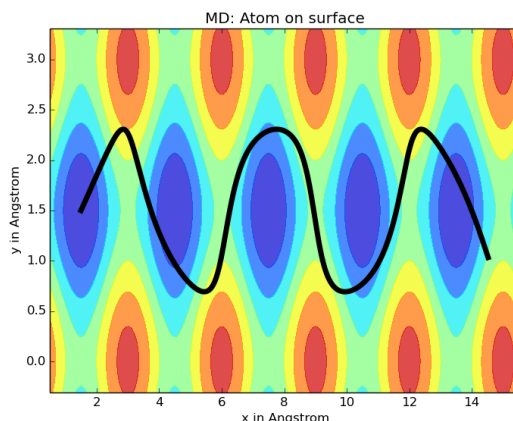
Bitte wenden!

In dieser Aufgabe sollen Sie Molekulardynamiksimulation der Bewegung eines Teilchen in diesem Potential durchführen. Dazu haben wir Ihnen ein Programm in einer weiteren Programmiersprache, nämlich Python, zur Verfügung gestellt. Python ist eine universelle höhere Programmiersprache, die in der Wissenschaftsgemeinde weit verbreitet ist, aber auch für andere Anwendungen beliebt ist.

Python ist für die meisten gängigen Betriebssysteme frei erhältlich und bei den meisten Linux-Distributionen im Standardumfang enthalten. Im Rahmen dieser Vorlesung empfehlen wir Ihnen, sich Python bei Anaconda , <https://docs.anaconda.com/> herunterzuladen. Dazu gibt es auch ein Einführungsvideo bei <https://www.youtube.com/watch?v=YJC6ldI3hWk&t=50s>. Diese Links sind auf der Moodle-Plattform zur Verfügung gestellt.

Das Python MD Programm `Diff_2D_MD_python.py` können Sie sich von der Webpage oder Moodle-Plattform der Vorlesung herunterladen. Sie sollen in dieser Aufgabe keine neue Programmiersprache lernen, öffnen Sie aber dennoch das Programm in einem Texteditor und schauen Sie es sich mal an. Nachdem Sie Python auf Ihrem Rechner installiert haben, können Sie das Programm mit `python Diff_2D_MD_python.py` starten.

Das Programm erlaubt es Ihnen, den *Verlet*-Algorithmus, den *Velocity Verlet*-Algorithmus und die *Generalisierte Langevin*-Gleichung zu benutzen. Wählen Sie zunächst eine der ersten beiden Methoden. Als nächstes werden Sie nach der Anzahl der Simulationsschritte gefragt, die der Simulationszeit in fs entspricht. Danach müssen Sie die kinetische Energie in eV eingeben. Die Simulationen werden im Potentialminimum bei $x = 1.5 \text{ \AA}$ und $y = 1.5 \text{ \AA}$ gestartet, allerdings müssen Sie die Anfangsrichtung in Grad eingeben. Nach der letzten Eingabe wird das Programm gestartet und nach der Ausführung die Trajektorie automatisch geplottet. Ein Beispiel sehen Sie hier:



Beachten Sie, dass je nach Verlauf der Trajektorie dabei die x - und y -Achsen unterschiedlich skaliert werden. Sie können die Trajektorie als .png Datei speichern. Nachdem Sie das Bild geschlossen haben, erscheint automatisch ein Diagramm mit der Gesamtenergie und der kinetischen Energie als Funktion der Simulationszeit. Zusätzlich wird die Datei `Output.dat` erzeugt, die den Zeitschritt, die potentielle, die kinetische und die Gesamt-Energie entlang der Trajektorie enthält.

Bei einer NVE Simulation, d.h. für konstante Gesamtenergie, erhalten Sie Trajektorien, die entweder auf eine Einheitszelle beschränkt bleiben oder die, wie im Bild oben, einer fort dauernden Propagation entsprechen. Laden Sie für beide Fälle ein Bild einer Beispieltrajektorie hoch. Welche Anfangsbedingungen entsprechen den beiden Fällen?

Bei einer realen Diffusion eines Teilchens auf einer Oberfläche tauscht das diffundierende Teilchen allerdings ständig Energie mit den Oberflächenatomen über Stöße aus. Dies kann man mit Hilfe der *Generalisierte Langevin*-Gleichung simulieren. Wählen Sie dazu nach Start des MD-Programms die Langevin-Methode und geben Sie dann als erstes die Temperatur in K ein, wobei Sie zunächst Raumtemperatur = 300 K eingeben sollten. Zusätzlich müssen Sie den Exponenten des Reibungskoeffizienten angeben, wobei 13, d.h. $\gamma = 10^{13} \text{ s}^{-1}$ eine gute Wahl ist. Führen Sie nun längere Simulation mit 50.000 Schritten durch. Was beobachten Sie? Laden Sie ein Bild einer typischen Trajektorie zusammen mit der Energie als Funktion der Laufzeit hoch. Versuchen Sie, aus der Datei `Output.dat` die mittlere potentielle, kinetische und Gesamt-Energie zu berechnen und bewerten Sie das Ergebnis.