

Serie 10

Best Before: 31.5/1.6, in den Übungsgruppen (2 wochen)

Koordinatoren: Alexander Dabrowski, HG G 52.1, alexander.dabrowski@sam.math.ethz.ch

Webpage: http://www.math.ethz.ch/education/bachelor/lectures/fs2016/other/nm_pc

1. Molekuldynamik

Wir betrachten ein System von N Teilchen, deren paarweise Wechselwirkung durch das Lennard-Jones Potential $U(r) = U(\|\underline{x}\|)$ gegeben ist:

$$U(r) = 4\epsilon \left(\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right) \quad (1)$$

und:

$$V(\underline{x}_1, \dots, \underline{x}_N) := \frac{1}{2} \sum_{i=1}^N \sum_{\substack{j=1 \\ i \neq j}}^N U(\|\underline{x}_i - \underline{x}_j\|) \quad (2)$$

Die Hamiltonfunktion des Systems ist dann:

$$\mathcal{H}(\underline{x}_1, \dots, \underline{x}_N, \underline{p}_1, \dots, \underline{p}_N) = \sum_{i=1}^N \frac{p_i^2}{2m} + V(\underline{x}_1, \dots, \underline{x}_N). \quad (3)$$

Verwenden Sie die Hamiltongleichungen zur Herleitung der Bewegungsgleichungen.

a) Implementieren Sie das Velocity-Verlet Verfahren (8).

b) Implementieren Sie das Leap-Frog Verfahren (9).

Hinweis (aus dem Skript): Aus den Hamiltongleichungen:

$$\begin{aligned} \dot{\underline{x}}_i &= \nabla_{\underline{p}_i} \mathcal{H} \\ \dot{\underline{p}}_i &= -\nabla_{\underline{x}_i} \mathcal{H} \end{aligned} \quad (4)$$

folgt

$$\begin{aligned} \dot{\underline{x}}_i &= \underline{v}_i \\ \dot{\underline{v}}_i &= -\nabla_{\underline{x}_i} V(\underline{x}_1, \dots, \underline{x}_N) \end{aligned} \quad (5)$$

oder

$$m\ddot{\underline{x}}_i = \underline{F}_i \quad (6)$$

Bitte wenden!

Die Ableitung in Gl. (6) approximieren wir mit finiten Differenzen 2. Ordnung.

$$m_i \frac{\underline{x}_i^{n+1} - 2\underline{x}_i^n + \underline{x}_i^{n-1}}{\delta t^2} = \underline{F}_i^n$$

$$\Rightarrow \underline{x}_i^{n+1} = 2\underline{x}_i^n - \underline{x}_i^{n-1} + \delta t^2 \underline{F}_i^n \quad (7)$$

Zur numerischen Lösung benutzen wir das Velocity-Verlet (8) und das Leap-Frog Verfahren (9).

$$\underline{v}_i^{n+1} = \underline{v}_i^n + \frac{\delta t}{2} (\underline{F}_i^n + \underline{F}_i^{n+1}) \quad (8)$$

$$\underline{x}_i^{n+1} = \underline{x}_i^n + \delta t \underline{v}_i^n + \frac{\delta t^2}{2} \underline{F}_i^n$$

$$\underline{v}_i^{n+\frac{1}{2}} = \underline{v}_i^{n-\frac{1}{2}} + \delta t \underline{F}_i^n \quad (9)$$

$$\underline{x}_i^{n+1} = \underline{x}_i^n + \delta t \underline{v}_i^{n+\frac{1}{2}}$$

Da beim Leap-Frog Verfahren die Geschwindigkeiten um $\frac{\delta t}{2}$ verschoben sind, müssen wir einmalig den Startwert anpassen:

$$\underline{v}_i^{-\frac{1}{2}} = \underline{v}_i^0 - \frac{\delta t}{2} \underline{F}_i^0 \quad (10)$$

Zur Berechnung der kinetischen Energie interpolieren wir die Geschwindigkeiten zu den Zeiten t_n linear:

$$\underline{v}_i^n = \frac{\underline{v}_i^{n+\frac{1}{2}} + \underline{v}_i^{n-\frac{1}{2}}}{2} \quad (11)$$

- c) Integrieren Sie das System von $t = [0, \dots, 60]$ mit $\delta t = 0.05$.

Hinweis: Die Funktion `init_pos_vel` aus `md.Template.py` erzeugt die Anfangswerte.

- d) Berechnen Sie die kinetische, potentielle und totale Energie für jeden Zeitschritt und erstellen Sie einen Plot.
- e) Was geschieht, wenn man einen gegebenen Anfangszustand mit einer minimalen Störung versieht?

Die Funktion `small_perturbation` aus dem Template simuliert zwei 25-Teilchen Systeme welche zu Beginn gleichmässig auf einem Gitter verteilt sind mit zufällig ausgewählten Geschwindigkeiten auf einem Kreis mit Radius 10^{-4} . Beim gestörten System wird bei einem einzigen Teilchen die Geschwindigkeit um 10^{-10} erhöht. Führe oben genannte Funktion aus und beschreibe deine Beobachtungen.

Warnung: Die Berechnungen können mehrere Minuten in Anspruch nehmen.

- f) Interpretieren und erklären Sie die einzelnen Plots aus der vorangehenden Unteraufgabe.

Siehe nächstes Blatt!

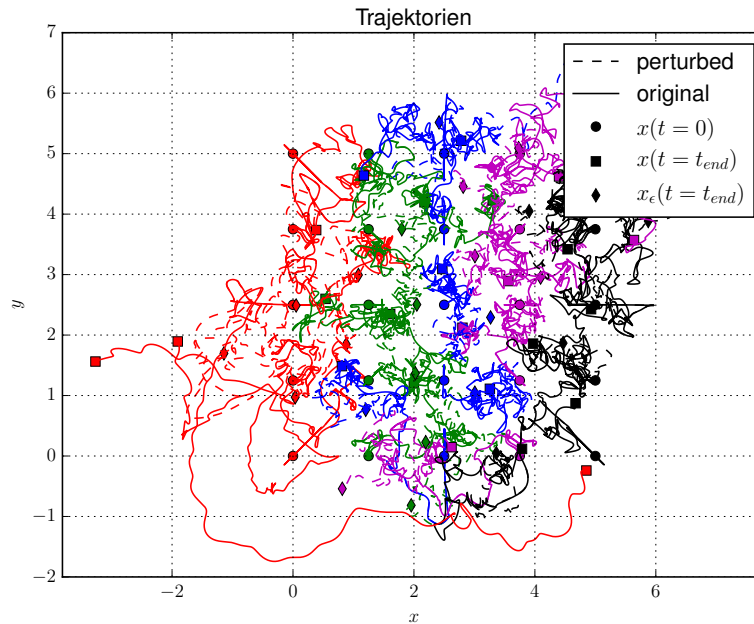


Abbildung 1: Aufgabe 1: Trajektorien (Zufällige Perturbation!)

2. Molekulardynamik II

Wir betrachten ein System von N Teilchen, deren paarweise Wechselwirkung durch das Lennard-Jones Potential $U(r) = U(\|x\|)$ gegeben ist:

$$U(r) = 4\epsilon \left(\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right) \quad (12)$$

und:

$$V(\underline{q}_1, \dots, \underline{q}_N) := \frac{1}{2} \sum_{i=1}^N \sum_{\substack{j=1 \\ i \neq j}}^N U(\|\underline{q}_i - \underline{q}_j\|) \quad (13)$$

Die Hamiltonfunktion des Systems ist dann:

$$\mathcal{H}(\underline{q}_1, \dots, \underline{q}_N, \underline{p}_1, \dots, \underline{p}_N) = \sum_{i=1}^N \frac{\underline{p}_i^2}{2m} + V(\underline{q}_1, \dots, \underline{q}_N). \quad (14)$$

Wir wollen die Zeitevolution mittels Splittingverfahren berechnen.

a) Leiten Sie die Splittingoperatoren Φ_a, Φ_b aus den Hamiltongleichungen:

$$\begin{aligned} \dot{\underline{q}}_i &= \nabla_{\underline{p}_i} \mathcal{H} \\ \dot{\underline{p}}_i &= -\nabla_{\underline{q}_i} \mathcal{H} \end{aligned} \quad (15)$$

analog dem Skript her.

b) Implementieren Sie diese Operatoren in den beiden Funktionen PhiT, PhiV.

Bitte wenden!

- c) Studieren Sie die Klasse `SplittingParameters` aus dem Skript und benutzen Sie ihre Funktion `intsplint` um die Zeitevolution zu berechnen. Verändern Sie den Code dieser Klasse nicht. Schreiben Sie `PhiV`, `PhiT` so, dass sie kompatibel sind.
- d) Verwenden Sie die Methoden `SS` (“Strang”) und `BM42` zur Zeitintegration. Plotten Sie die Erhaltungsgrößen.

3. Kernaufgabe: Exponentielles Euler-Verfahren (Prüfungsaufgabe FS14)

Problemstellung

Betrachten Sie das *exponentielle Euler-Verfahren* mit konstanter Schrittweite:

$$\underline{y}_{k+1} = \underline{y}_k + h\varphi(h\mathbf{J}_f)\mathbf{f}(\underline{y}_k), \quad k = 0, \dots, N \quad (16)$$

wobei:

$$\mathbf{J}_f := D\mathbf{f}(\underline{y}_k), \quad \varphi(z) = \frac{e^z - 1}{z}$$

Aufgabenstellung

- a) Leiten Sie die Stabilitätsfunktion $S(z)$ von (16) her.
- b) Schreiben Sie eine Python-Funktion `expEV` die das nichtlineare Anfangswertproblem

$$\dot{\underline{y}} = \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} -\frac{y_1^2}{y_2} + y_2 \log(y_2) \\ -y_1 \end{bmatrix}, \quad \underline{y}(0) = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

mit dem exponentiellen Eulerverfahren (16) mit konstanter Schrittweite löst.

Hinweis: Verwenden Sie das Template `exp_euler_Template.py`.

- c) Bestimmen Sie empirisch die Konvergenzordnung des Verfahrens. Betrachten Sie das Zeitintervall $[0, 6]$ und berechnen Sie den Fehler bezüglich der exakten Lösung:

$$\underline{y}(t) = \begin{bmatrix} -\cos(t) \exp(\sin(t)) \\ \exp(\sin(t)) \end{bmatrix}$$

für verschiedene Anzahl von Zeitschritten $N = 24, 48, 96, 192, 384$.