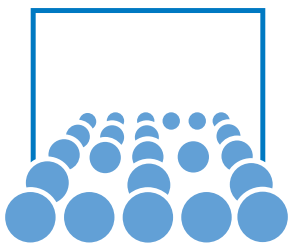
PSE Molekulardynamik - Entwicklung eines Molekulardynamik-Simulators

Molekulardynamik-Simulation: Algorithmus

Wolfgang Eckhardt

21. Oktober 2011



Übersicht

MD-Simulation: Prinzipieller Ablauf

Zeitintegration

Kraftbestimmung

C++ vs. C / Java

MD-Simulation: Prinzipieller Ablauf

- Gegeben: Moleküle mit Positionen x und Geschwindigkeiten v (sog. Phasenraum)
- Zwischen Molekülen wirken Kräfte
- Kraft auf ein Molekül i:

$$F_i = \sum_{i \neq j} F_{ij}$$

- Molekül wird gemäß diesen Kräften bewegt, d.h. eine neue Position und neue Geschwindigkeit werden bestimmt

$$F = m \cdot \ddot{x} \Rightarrow \ddot{x} = \frac{F}{m}$$

.

- Gewöhnliche Differentialgleichung mit Anfangsbedingungen: $F = m\ddot{x}$, gesucht x zu einem Zeitpunkt t.
- Wie bestimmt man die Kräfte?
- Wie bestimmt man Position x und Geschwindigkeit v?

Diskretisierung - Zeitintegration

$$F = m \cdot \ddot{x}$$

Diskretisierung - Zeitintegration

$$F = m \cdot \ddot{x}$$

- Diskretisierung der Ableitungen durch Differenzenquotienten (aus Taylor-Reihe hergeleitet):

$$\frac{\delta x}{\delta t} = \frac{x(t+\delta t) - x(t-\delta t)}{2 \cdot \delta t} \tag{1}$$

$$= \frac{x^{n+1} - x^{n-1}}{2 \cdot \delta t} \tag{2}$$

$$\frac{\delta^2 x}{\delta t^2} = \frac{x^{n+1} - 2 \cdot x^n + x^{n-1}}{\delta t^2} \tag{3}$$

- Einsetzen in unsere DGL liefert

$$F_i^n = m \cdot \ddot{x}_i^n \tag{4}$$

$$= m \cdot \frac{x_i^{n+1} - 2 \cdot x_i^n + x_i^{n-1}}{\delta t^2} \tag{5}$$

$$x_i^{n+1} = 2 \cdot x_i^n - x_i^{n-1} + \delta t^2 \cdot F_i^n / m_i \tag{6}$$

Diskretisierung - Zeitintegration

- Störmer-Verlet-Verfahren:

$$x_i^{n+1} = 2 \cdot x_i^n - x_i^{n-1} + \delta t^2 \cdot F_i^n / m_i \quad (7)$$

$$v_i^n = \frac{x^{n+1} - x^{n-1}}{2 \cdot \delta t} \quad (8)$$

- Velocity-Störmer-Verlet-Verfahren:
  - (8) nach x^{n-1} auflösen und in (7) einsetzen liefert (9)
  - weitere Umformungen von (7) und (8) ergeben (10):

$$x_i^{n+1} = x_i^n + \delta t \cdot v_i^n + \frac{F_i^n \cdot \delta t^2}{2 \cdot m_i} \quad (9)$$

$$v_i^{n+1} = v_i^n + \frac{(F^n + F^{n+1})\delta t}{2 \cdot m_i} \quad (10)$$

Kraftbestimmung

- Potential U beschreibt potentielle Energie eines Körpers
- Kraft $F = -\nabla U$
- Bsp.: Gewichtskraft / Lageenergie
- Für erste Aufgabe:
  - (skaliertes) Gravitationspotential

$$U(r_{ij}) = -G_{Grav} \cdot \frac{m_i m_j}{r_{ij}} \quad \texttt{mit} \quad G_{Grav} = 1,$$

$$r_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}$$

  - mit der resultierenden Kraft

$$F(r_{ij}) = \frac{m_i m_j}{r_{ij}^3} \cdot \vec{r_{ij}}$$

- Kraft über alle Molekülpaare aufsummieren:

$$F_i = \sum_{i \neq j} F_{ij}$$

C++ vs. C / Java

- Klassendefinition / -Deklaration
- Trennung Header (.h) / Implementierung (.cpp)
- Vererbung (public */ protected / private*) */ Mehrfachvererbung*
- Methoden per Default nicht überschreibbar (explizite Kennzeichnung mit **virtual**)
- Pointer vs. Referenzen vs. Werte (Copy-Konstruktoren)

Debugging:

- gdb (Tutorial: `http://www.cs.cmu.edu/~gilpin/tutorial/`)
- Valgrind (Memcheck)