

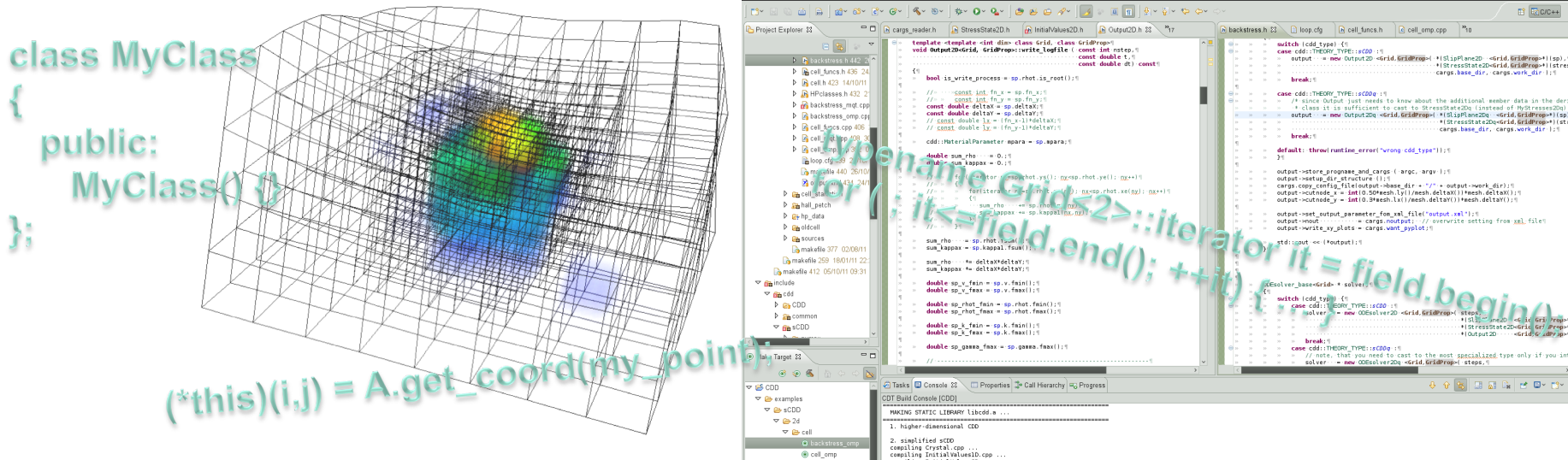
Wissenschaftliches Programmieren für Ingenieure

T. Achkar, A. Trenkle, Dr. M. Stricker, Dr. D. Weygand

Übung 8 Molekulardynamik (MD) C++ 11

Institute for Applied Materials – Computational Materials Science (IAM-CMS)

KIT – The Karlsruhe Institute of Technology – University of the State of Baden-Wuerttemberg, GERMANY



Info

- Im Archive ws_II_WS1617.tgz befindet sich der Quellcode für ein 2d MD Programm(siehe auch Vorlesungsfolien) unter Verwendung von Nachbarschaftslisten. Als Anfangsstruktur wird entweder ein hexagonales- oder ein quadratisches Gitter gesetzt. Die jeweiligen Konfigurationsdateien finden Sie in den **Unterordnern**:
 - Ordner hex: Hexagonales Gitter
 - Ordner quad: Quadratisches Gitter
- Dateiformat:
 - 1. Zeile: Anzahl der Teilchen
 - Für jedes Teilchen eine Zeile: no x y vx vy
- Die Datei „**start.dat**“ (in den Unterordnern) enthält Anfangswerte zum Starten des Programms
- Dateiformat:
 - 1. Zeile: Dateiname für Anfangsstruktur
 - 2. Zeile: Größe des Simulationsgebiets (es müssen alle Atome, die in der Datei von Zeile 1 aufgeführt sind, in das Gebiet passen; beim Quadratgitter sind die Abstände zwischen den nächsten Nachbarn 1; beim Hexagonalgitter sind die Abstände 1 in der Reihe und $\sqrt{3}/2$ zwischen den Reihen)
 - 3. Zeile: Offset (x,y): Alle Positionen von der Datei (Zeile 1) werden um (x,y) verschoben (default (0,0) == keine Verschiebung)
 - 4. Zeile: „nahe“ Gleichgewichtsabstand == Gitterkonstante [sigma]
 - 5. Zeile: Wert des Abscheideradius [sigma]
 - 6. Zeile: Gesamtzeit (normalisierten Einheiten- siehe Programmcode)
 - 7. Zeile: dt Zeitschritt
 - 8. Zeile: Integrationsverfahren (1: Euler; 2: Verlet einfach; 3: Verlet mit Nachbarschaftslisten)

Todo

- Das Programm mit Makefile übersetzen

- In den Unterordnern:

- Programm aufrufen: `../mdsim < start.dat`
- Dateien nacheinander in gnuplot anzeigen:

```
do for [ii=0:1000:1] {file =sprintf('lattice%d.dat',ii); pl file u 2:3 title file}
```

- Beobachten Sie Unterschiede in der Stabilität der Ausgangskonfigurationen?
- Verändern Sie die Ausgangsgitterkonstante (1.12 um +/-20%)
 - Erstellen Sie hierzu ein Skript!!
 - Welche Unterschiede beobachten Sie?

- Verändern Sie das Integrationsverfahren Euler/Verlet (Zeile 8 in start.dat). Vergleichen Sie die Teilchenpositionen des letzten Schrittes der beiden Verfahren.
- Messen Sie die Rechenzeit der Optionen 2 (Verlet) und 3 (Verlet mit Nachbarschaftslisten) z.B. mit dem Konsolenbefehl time

```
zbsdbook:mandelbrot daniel$ time ./a.out
pixel: 135000

real    1m59.141s
user    1m58.822s
sys      0m0.244s
```

real: verstrichene Zeit
user: CPU - Zeit (bei Verwendung mehrerer Threads ist diese Zeit größer als die verstrichene Zeit)
Sys: CPU Zeit, die vom System verbraucht wurde (z.Bsp für Ein/Ausgabe etc)

- Identifizieren Sie die rechnerzeitintensiven Teile des Programms
 - Beschleunigen Sie diese Teile mit Hilfe von OpenMP Anweisungen.
 - Messen Sie danach die Rechenzeit (siehe letzte Übung).