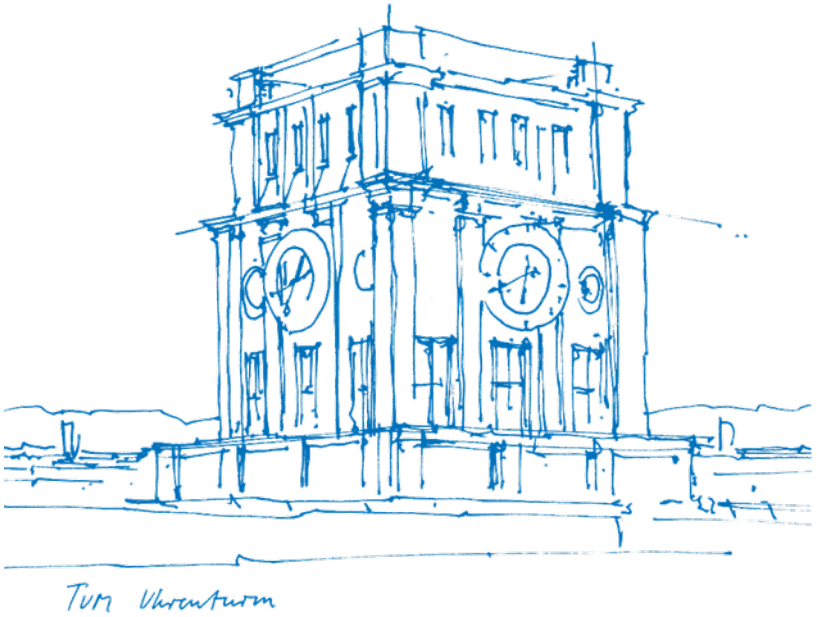
Molecular Dynamics - Assignment 2

Alex Hocks Jan Hampe Johannes Riemenschneider
Technische Universität München
TUM CIT
Lehrstuhl für wissenschaftliches Rechnen
17. November 2022



Example of new input file

```
#Random comments
# xyz-coord      velocity        mass       shape       dimensions      distance
2
0.0 0.0 0.0      0.0 0.0 0.0     1.0
7.0 7.0 7.0      1.0 1.0 1.0     2.0e-8     CuboiD      4        5        6         3.0
# epsilon    sigma
0.1         0.2
```

Structure:

- Comments
- Number of bodies

Example of new input file

```
#Random comments
# xyz-coord      velocity        mass       shape       dimensions      distance
2
0.0 0.0 0.0      0.0 0.0 0.0     1.0
7.0 7.0 7.0      1.0 1.0 1.0     2.0e-8     CuboiD      4        5        6         3.0
# epsilon    sigma
0.1         0.2
```

Structure:

- Comments
- Number of bodies
- Bodies (with Shape, Dimensions and distance as optional)

Example of new input file

```
#Random comments
# xyz-coord      velocity        mass       shape       dimensions      distance
2
0.0 0.0 0.0      0.0 0.0 0.0     1.0
7.0 7.0 7.0      1.0 1.0 1.0     2.0e-8     CuboiD      4        5        6         3.0
# epsilon    sigma
0.1         0.2
```

Structure:

- Comments
- Number of bodies
- Bodies (with Shape, Dimensions and distance as optional)
- Comments

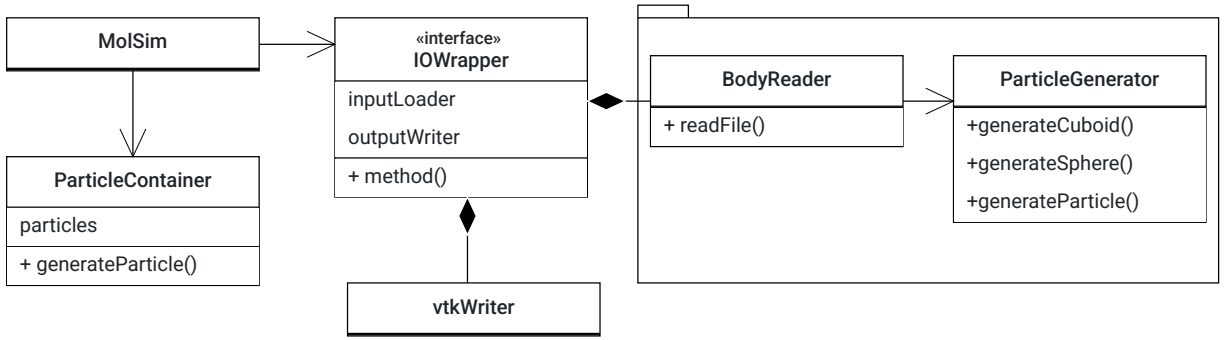
Example of new input file

```
#Random comments
# xyz-coord      velocity        mass       shape       dimensions      distance
2
0.0 0.0 0.0      0.0 0.0 0.0     1.0
7.0 7.0 7.0      1.0 1.0 1.0     2.0e-8     CuboiD      4        5        6         3.0
# epsilon    sigma
0.1         0.2
```

Structure:

- Comments
- Number of bodies
- Bodies (with Shape, Dimensions and distance as optional)
- Comments
- Definition of epsilon and sigma (optional)

IO



input Loader gets chosen at compile time

Input parsing - Definition of Body

```
enum Shape {cuboid, sphere};

struct Body {
        Shape shape;
        Eigen::Vector3d fixpoint;
        Eigen::Vector3d dimensions;
        double distance;
        double mass;
        Eigen::Vector3d start_velocity;
};
```

CI/CD

- Protection of master branch

CI/CD

- Protection of master branch
- Deployment of CI/CD pipeline for *all* branches

CI/CD

- Protection of master branch
- Deployment of CI/CD pipeline for *all* branches

The pipeline consist of:

- library installation

CI/CD

- Protection of master branch
- Deployment of CI/CD pipeline for *all* branches

The pipeline consist of:

- library installation
- build process

CI/CD

- Protection of master branch
- Deployment of CI/CD pipeline for *all* branches

The pipeline consist of:

- library installation
- build process
- sanitizers

CI/CD

- Protection of master branch
- Deployment of CI/CD pipeline for *all* branches

The pipeline consist of:

- library installation
- build process
- sanitizers
- unit tests for every major component

Logging

format:

```
[time][level::context] message

[19:27:28][info::simulation] starting simulation
```

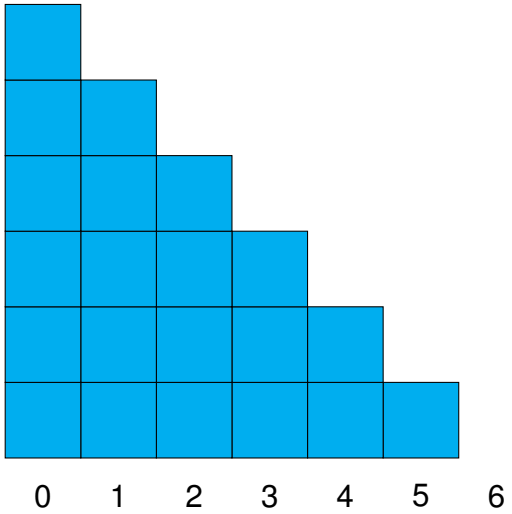
- 6 different log-levels available
- log-level can be chosen via command line input

Performance optimization

Force calculation is the only part relevant to performance

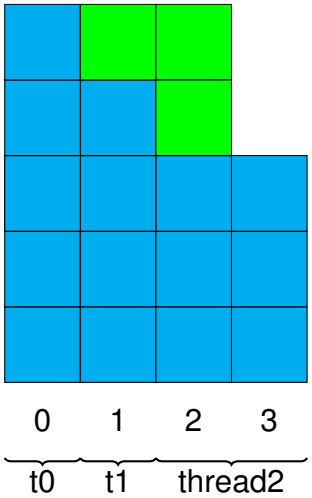
```
    void calculateFLennardJones() {
        //set all current forces on all particles to 0
        particleContainer.forAllParticles([](Particle &p) {
                p.setOldF(p.getF());
                p.setF({0., 0., 0.});
        });

        particleContainer.forAllPairs([](Particle &p1, Particle &p2){
                calculate force

                p1.add_to_F(force);
                p2.add_to_F(-force);
        });
}
```

Performance optimization - "Gaussian multithreading"



- Idea: Force calculation can be multithreaded quite easily
- Evenly distribute Particle-pairs among multiple threads
- One rectangle represents one necessary force-calculation where $\min(p1, p2)$ is the number displayed below

Performance optimization - "Gaussian multithreading"



- Create "Gaussian rectangle“ as good as possible and distribute the resulting blocks
- Threads use personal accumulators
- Accumulators get added in the end

In the end outsourcing the problem to OpenMP turned out to be much easier

Performance optimization - Refactoring of ParticleContainer and SIMD

- Utilization of seperate vectors for forces, velocities and positions instead of using one struct

Performance optimization - Refactoring of ParticleContainer and SIMD

- Utilization of seperate vectors for forces, velocities and positions instead of using one struct
- Flattening of the two-dimensional Vectors:
  std::vector<Eigen::Vector3d> $\rightarrow$ std::vector<double>

Performance optimization - Refactoring of ParticleContainer and SIMD

- Utilization of seperate vectors for forces, velocities and positions instead of using one struct
- Flattening of the two-dimensional Vectors:
  std::vector<Eigen::Vector3d> $\rightarrow$ std::vector<double>

  $\longrightarrow$ Perfect setup for SIMD-Instructions (no working version implemented yet)(with 32 byte alignment)

The OpenMP-approach

Roadblocks

Spontaneously imploding Laptops and the lessons they teach us

- Get yourself familiar with alternative options available to not completely disrupt your workflow

Roadblocks

Spontaneously imploding Laptops and the lessons they teach us

- Get yourself familiar with alternative options available to not completely disrupt your workflow
- Create backups!

Roadblocks

Spontaneously imploding Laptops and the lessons they teach us

- Get yourself familiar with alternative options available to not completely disrupt your workflow
- Create backups!
- People are nice, ask for help!

Roadblocks

Spontaneously imploding Laptops and the lessons they teach us

- Get yourself familiar with alternative options available to not completely disrupt your workflow
- Create backups!
- People are nice, ask for help!

Other Roadblocks

- Missing overview over the already existing functionality and helper functions

Roadblocks

Spontaneously imploding Laptops and the lessons they teach us

- Get yourself familiar with alternative options available to not completely disrupt your workflow
- Create backups!
- People are nice, ask for help!

Other Roadblocks

- Missing overview over the already existing functionality and helper functions
- Missing communication on interfaces, project structure, etc.

Roadblocks

Spontaneously imploding Laptops and the lessons they teach us

- Get yourself familiar with alternative options available to not completely disrupt your workflow
- Create backups!
- People are nice, ask for help!

Other Roadblocks

- Missing overview over the already existing functionality and helper functions
- Missing communication on interfaces, project structure, etc.
- Documentation and Code examples are your friend