

# Week 1 Molecular Dynamics

Alex Hocks   Jan Hampe   Johannes Riemenschneider

Technische Universität München

TUM CIT

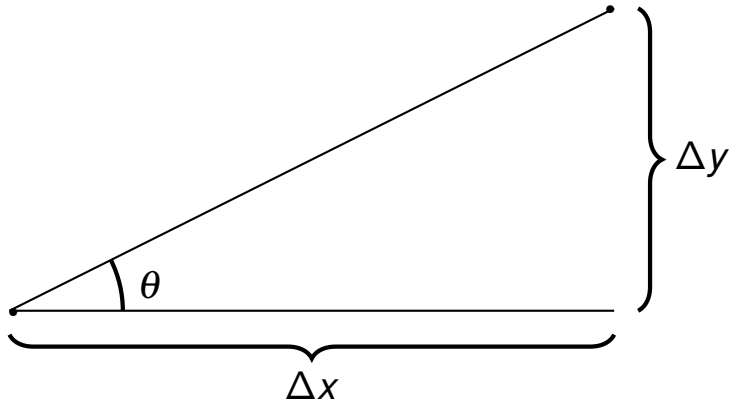
Lehrstuhl für wissenschaftliches Rechnen

3. November 2022



*TUM Uhrenturm*

## 2 Dimensional Force Calculation



$$\Delta x = x_2 - x_1 \quad (1)$$

$$\Delta y = y_2 - y_1 \quad (2)$$

$$|F| = \frac{m_1 m_2}{\Delta x^2 + \Delta y^2} \quad (3)$$

$$F_x = \cos(\theta) \cdot |F| = \Delta x \cdot \frac{m_1 m_2}{(\Delta x^2 + \Delta y^2)^{3/2}} \quad (4)$$

$$F_y = \sin(\theta) \cdot |F| = \Delta y \cdot \frac{m_1 m_2}{(\Delta x^2 + \Delta y^2)^{3/2}} \quad (5)$$

# Utilizing $F_{ij} = -F_{ji}$

The naive approach ( $n \cdot (n - 1)$  Force calculations):

```
for all Particles p:
    for all Particles p'!=p:
        computeF(p,p')
```

•

•

•



# Utilizing $F_{ij} = -F_{ji}$

The naive approach ( $n \cdot (n - 1)$  Force calculations):

```
for all Particles p:
    for all Particles p'!=p:
        computeF(p,p')
```



# Utilizing $F_{ij} = -F_{ji}$

The naive approach ( $n \cdot (n - 1)$  Force calculations):

```
for all Particles p:
    for all Particles p'!=p:
        computeF(p,p')
```

•



•

•

# Utilizing $F_{ij} = -F_{ji}$

The naive approach ( $n \cdot (n - 1)$  Force calculations):

```
for all Particles p:
    for all Particles p'!=p:
        computeF(p,p')
```



# Utilizing $F_{ij} = -F_{ji}$

A better approach ( $\frac{1}{2}n \cdot (n-1)$  Force calculations):

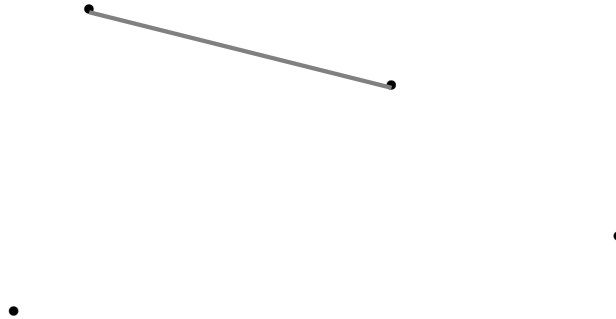
```
for all ParticlePairs (p,p'):
    computeF(p,p')
```



# Utilizing $F_{ij} = -F_{ji}$

A better approach ( $\frac{1}{2}n \cdot (n - 1)$  Force calculations):

```
for all ParticlePairs (p,p'):
    computeF(p,p')
```

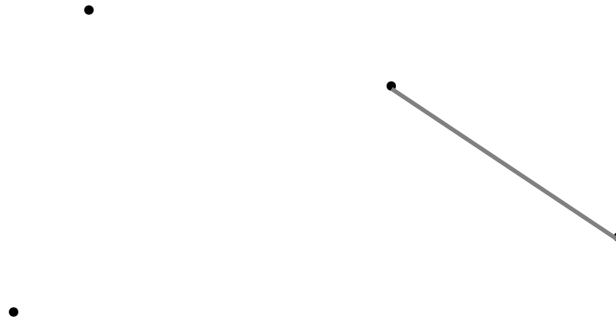




# Utilizing $F_{ij} = -F_{ji}$

A better approach ( $\frac{1}{2}n \cdot (n-1)$  Force calculations):

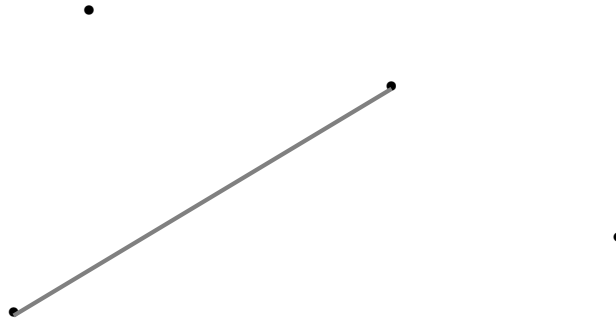
```
for all ParticlePairs (p,p'):
    computeF(p,p')
```



# Utilizing $F_{ij} = -F_{ji}$

A better approach ( $\frac{1}{2}n \cdot (n - 1)$  Force calculations):

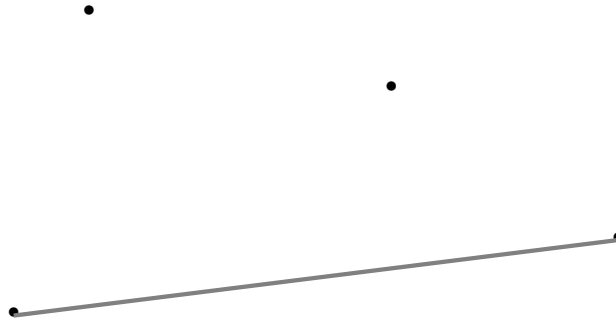
```
for all ParticlePairs (p,p'):
    computeF(p,p')
```



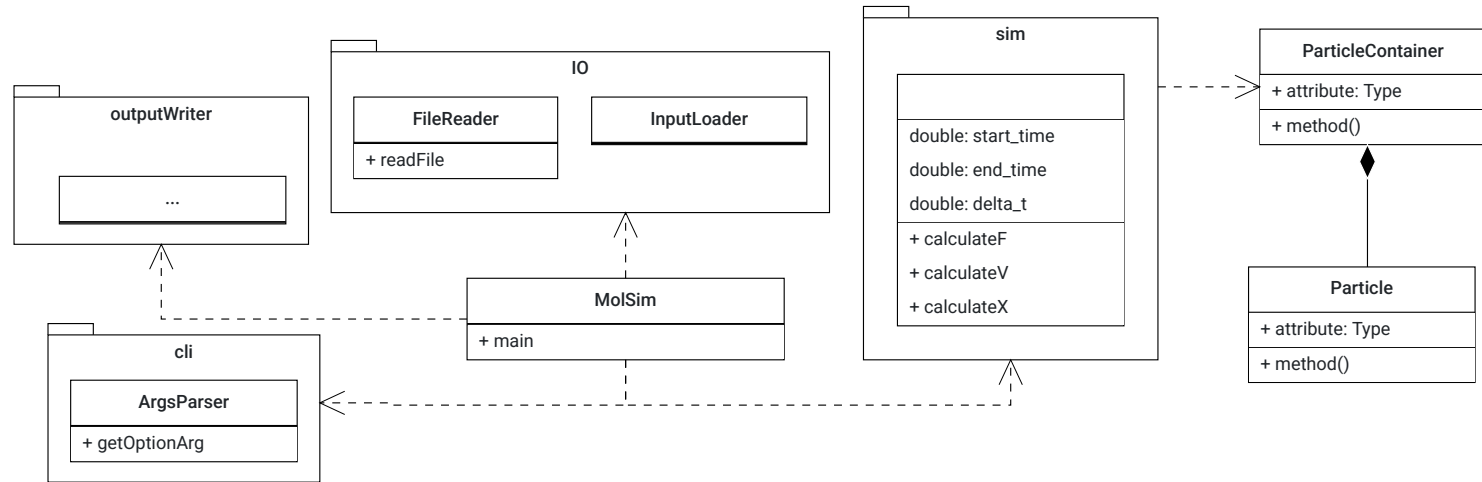
# Utilizing $F_{ij} = -F_{ji}$

A better approach ( $\frac{1}{2}n \cdot (n - 1)$  Force calculations):

```
for all ParticlePairs (p,p'):
    computeF(p,p')
```



# Refactoring



# Use of the Program

```
> ./MolSim --help
```

```
Welcome to MolSim Help
```

```
Usage:
```

```
MolSim <input-files> <options>
```

```
Options:
```

<code>--help, -h</code>	Prints this screen.
<code>-dt &lt;value&gt;</code>	Sets delta time to <value>. If -dt is not specified default value is used.
<code>-et &lt;value&gt;</code>	Set end time to <value>. If -et is not specified default value is used.
<code>-o &lt;name&gt;</code>	Set base name of output files. DO NOT USE A PATH! Default is 'result'.
<code>-of &lt;path&gt;</code>	Set path to output folder. Default is ./output

# Generic IO

```
template <typename LOCATOR, void (*LOAD)(LOCATOR, std::list<Particle>&)>
class InputLoader {
    private:
        std::list<Particle> buffer;
        LOCATOR locator;
    public:
        explicit InputLoader(LOCATOR loc) : locator(loc) {}
        InputLoader(const InputLoader& i) = delete;
        void reload() { LOAD(locator, buffer); }
        void getParticles(std::vector<Particle>&buf) {...}
};
```

# Refactoring surprises

- Task: Verify the correctness of the program after major refactoring

# Refactoring surprises

- Task: Verify the correctness of the program after major refactoring
- Idea: string-compare the new output with the old output, if the program works as intended it should be the same



# Refactoring surprises

- Task: Verify the correctness of the program after major refactoring
- Idea: string-compare the new output with the old output, if the program works as intended it should be the same
- Observation: Minuscule differences in the outputs generated (unobservable in the videos)

# Refactoring surprises

- Task: Verify the correctness of the program after major refactoring
- Idea: string-compare the new output with the old output, if the program works as intended it should be the same
- Observation: Minuscule differences in the outputs generated (unobservable in the videos)

What happened:

# Refactoring surprises

- Task: Verify the correctness of the program after major refactoring
- Idea: string-compare the new output with the old output, if the program works as intended it should be the same
- Observation: Minuscule differences in the outputs generated (unobservable in the videos)

What happened:

- Floating point operations are not associative  $\rightarrow$  both outputs were “correct”

# Refactoring surprises

- Task: Verify the correctness of the program after major refactoring
- Idea: string-compare the new output with the old output, if the program works as intended it should be the same
- Observation: Minuscule differences in the outputs generated (unobservable in the videos)

What happened:

- Floating point operations are not associative  $\rightarrow$  both outputs were “correct”
- Probably some compiler magic

# Refactoring surprises

- Task: Verify the correctness of the program after major refactoring
- Idea: string-compare the new output with the old output, if the program works as intended it should be the same
- Observation: Minuscule differences in the outputs generated (unobservable in the videos)

What happened:

- Floating point operations are not associative  $\rightarrow$  both outputs were “correct”
- Probably some compiler magic
- First-hand encounter with the inaccuracies of numerical programming

bloopers

A large, bold, black-outlined rectangle containing the word "Click!" in a very large, black, sans-serif font.

Click!

# Ideas, Roadblocks and Pain

- Custom CSS Doxygen didn't cooperate

# Ideas, Roadblocks and Pain

- Custom CSS Doxygen didn't cooperate
- Disproportion of time invested  $\leftrightarrow$  relevance



# Ideas, Roadblocks and Pain

- Custom CSS Doxygen didn't cooperate
- Disproportion of time invested  $\leftrightarrow$  relevance
- libxerces, Eigen deployment pain

# Ideas, Roadblocks and Pain

- Custom CSS Doxygen didn't cooperate
- Disproportion of time invested  $\leftrightarrow$  relevance
- libxerces, Eigen deployment pain
- Cmake, Doctest details

# Ideas, Roadblocks and Pain

- Custom CSS Doxygen didn't cooperate
- Disproportion of time invested  $\leftrightarrow$  relevance
- libxerces, Eigen deployment pain
- Cmake, Doctest details
- Iterator-related debate

# Ideas, Roadblocks and Pain

- Custom CSS Doxygen didn't cooperate
- Disproportion of time invested  $\leftrightarrow$  relevance
- libxerces, Eigen deployment pain
- Cmake, Doctest details
- Iterator-related debate
- Data structure locality (for later exercises)