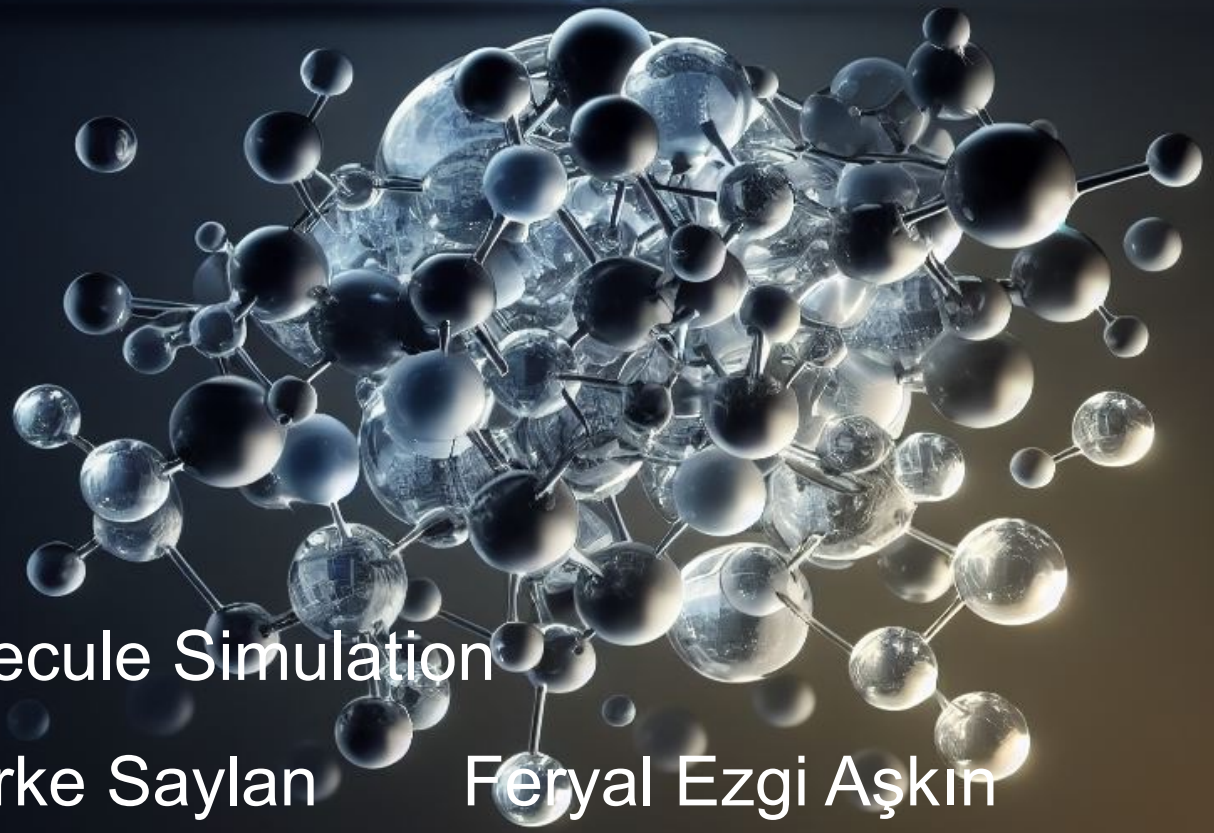
Bachelor-Praktikum - Scientific Computing (PSE)
Molecular Dynamics



Worksheet 2 – Molecule Simulation

Group F: Alp Kaan Aksu Berke Saylan Feryal Ezgi Aşkın

17.11.2023

Changes in Input File Processing

```
"simulation": {
  "model": "lennard_jones",
  "end_time": 5,
  "time_delta": 0.0002,
  "video_duration": 60,
  "frame_rate": 24,
  "output_type": "vtk",
  "output_path": "output",
  "sigma": 1,
  "epsilon": 5
},
"objects": [
  {
    "type": "cuboid",
    "type_id": 1,
    "position": [0, 0, 0],
    "velocity": [0, 0, 0],
    "size": [40, 8, 1],
    "mesh_width": 1.1225,
    "mass": 1
  },
```

```
Simulation Parameters
lennard_jones
5
0.0002
60
24
vtk
output
1
5
```

```
Cuboid 1 Parameters
cuboid
1
0.0 0.0 0.0
0 0 0
1.1225
1.0
0.4 0.0 0.0
0.0
0
```

Decision: We have shifted from using a .txt file to a .json file for specifying simulation input parameters!?

.json: Enhancing Simplicity and Flexibility

Reasoning: .json offers a more structured and flexible format, allowing for easy organization of complex data.

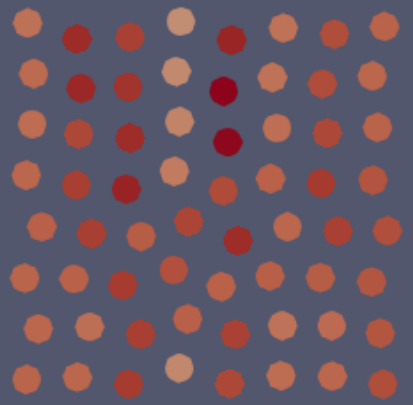
- Simplicity: Streamlining parameter specification, enhancing readability and maintainability
- Clarity: Hierarchical structure of .json improves parameter readability
- Flexibility: Easy accomodation of diverse data types, nested structures, and evolving simulation requirements

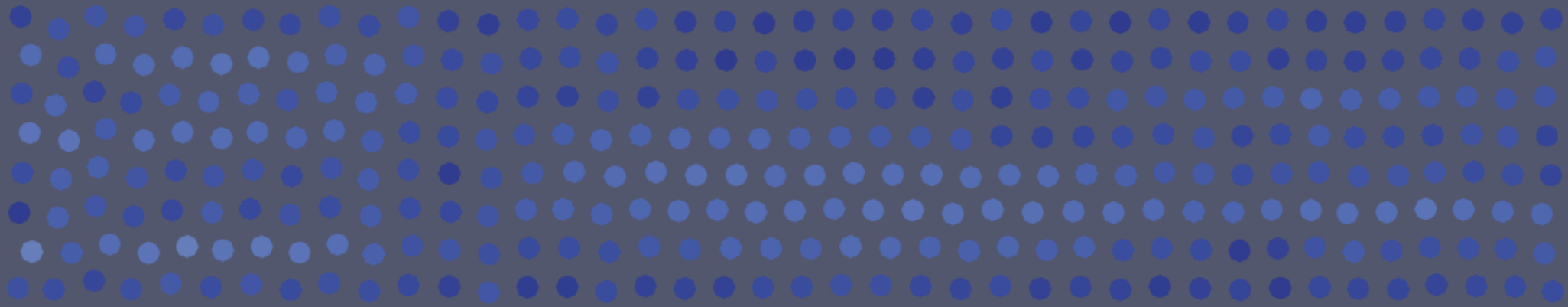
Brownian Motion in Simulation

- Particles exhibit continuous motion driven by thermal energy a.k.a Brownian motion
- Randomness leads to unpredictable fluctuations in particle direction and speed
- Randomness represented by probability distributions, e.g. the Maxwell-Boltzmann distribution

Visualization of Brownian Motion in Paraview

Different shades of the colors represent the velocity of the molecules





Problems encountered

- CMakeList.txt configuration for the integration of tests to the project

→ target_link_libraries(main-executable source-files) necessary

- We implemented the functionality for the Particle Generator only in a header instead of writing it in a .cpp file, despite knowing it's bad practice

→ Easier inclusion and sharing of functionality

→ However, caused: circular dependency

Definition: Mutual reliance between two headers

Compiler struggles to resolve dependencies

→ May result in errors or unexpected behavior

We now know why it might be a better idea to avoid header-only implementations.

Problems encountered

- We were confused about the direction of the position difference vector

$$F_{ij} = -\frac{\partial}{\partial x} U_{ij} \cdot \frac{x_i - x_j}{||x_i - x_j||_2}$$

$$F_{ij} = \frac{m_i m_j}{(||x_i - x_j||_2)^3}(x_j - x_i)$$

It’s more understandable when two formulas are put together but just looking at the code confused us for a second and produced a weird simulation (our cuboids exploded!) ☺

Problems encountered

- We were using a carriage return (‘\r’) to show a nice program output, where we update the percentage after each iteration.
- We tried to do the same using spdlog, but we couldn’t get it to work.

- Go back to std::cout?

Particle Generator

- Approach so far: Generating molecules individually
- Improvement: Particle Generator Class, so that particles are arranged in a 3D rectangular grid

- Position, Size, Mesh Width, Initial Velocity, Mass, Type ID etc. define cuboid properties

⟶ Streamlined particle generation, enhanced simulation capabilities

What will happen if we play with different parameters?

Lennard Jones Model

- Apparently only the formula for force calculation changes, so our model wrapper class from previous iteration was overkill
- Definitely more interesting than the previous model
- We calculated pow(distance, 6) and pow(sigma, 6) in advance to make the code more ‘efficient’, endless mathematical/code tricks to make it ‘efficient’

- epsilon: the depth of the potential well, which reflects the strength of the interaction between the particles
- sigma: distance at which the inter-particle potential is zero.

References

All images are generated by us using Microsoft Bing AI, so no copyright