Scientific Computing (PSE) Molecular Dynamics
Group D

Worksheet 5

02

SIMULATION TASKS

Task 1 to 3 -Worksheet 5

Simulation of a membrane

Storing neighboring particles and avoiding self-penetration to simulate a flexible membrane

Parallelization

Using the OpenMP framework to allow multithreading and parallel computation

Rayleigh-Taylor instability in 3D

Fluid cuboid with higher density above fluid cuboid with lower density subjected to gravitational forces in three dimensional space

TIMELINE

Parallelization & Rayleigh Taylor 3D

Parallel processing at the beginning to be able to have better performance (at least in some cases) in further tasks and simulations

Adaptation of Thermostat
Implementation of Nanotubes

Nanotubes with different fixed walls can be created with XML input. Extended thermostat to ignore total velocity of the fluid

Membrane

Kept the membrane as last task as it can work independently and seemed to be more complex to implement to us

We decided to do the membrane at the end and the “enhancing”, so feature addition or extension as well as performance enhancing through parallelization, at the beginning to use and test it in the different scenarios after that.

04

NANOSCALE FLOW SIMULATION

Task 4 - Option A

Fixed walls

Position-fixed cuoboids that represent walls but still interact (with Lennard-Jones forces) with the fluid inside the tube

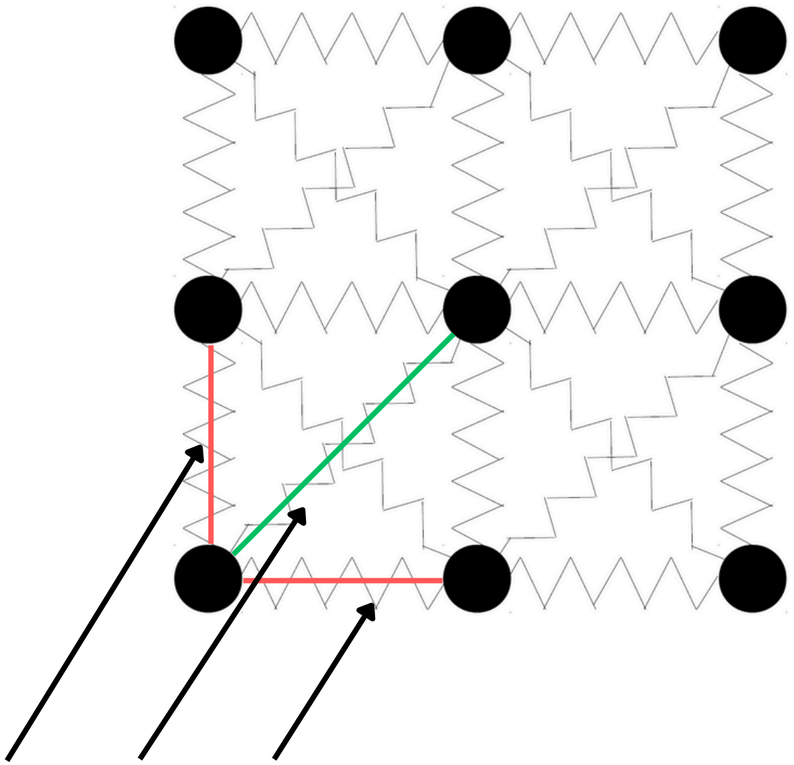
Adapted thermostat

Extension of the existing thermostat from last sheet, ignoring the total velocity of the fluid

Density & Velocity-Profile

Output of .csv files that give information about the density and velocity of molecules in each bin

Particle Class



Extended the Particle class to store neighbor molecules

Initialized on a rectangular grid, setting direct and diagonal neighbors

MEMBRANE

Features

Neighboring molecules with different properties than just two molecules next to each other

Avoidance of self penetration by repulsive Lennard-Jones

Neighboring particles interact in a harmonic potential (new force)

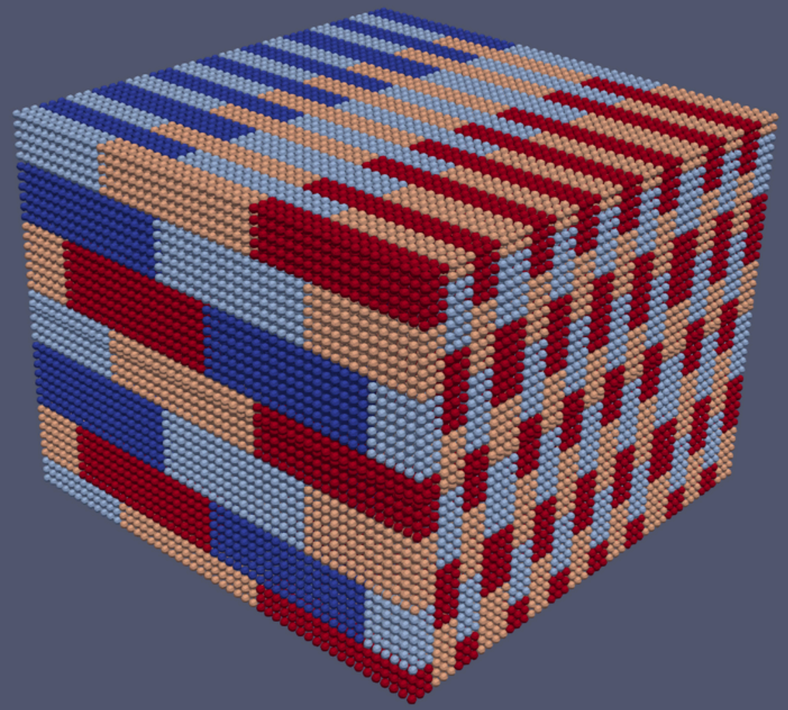
OpenMP


```
#pragma omp declare simd simdlen(4) uniform(this, x1, y1, z1, mass1, type1, correction) linear(ref(x2, y2, z2, mass2, type2))
#pragma omp declare simd simdlen(8) uniform(this, x1, y1, z1, mass1, type1, correction) linear(ref(x2, y2, z2, mass2, type2))
  virtual void inline calculateForce_3D(
      double const &x1,
      double const &y1,
      double const &z1,
      double const &mass1,
      long const &type1,
      double &x2,
      double &y2,
      double &z2,
      double &mass2,
      long &type2,
      double &result_x,
      double &result_y,
      double &result_z,
      std::array<double, 3> &correction) = 0;
```

simd

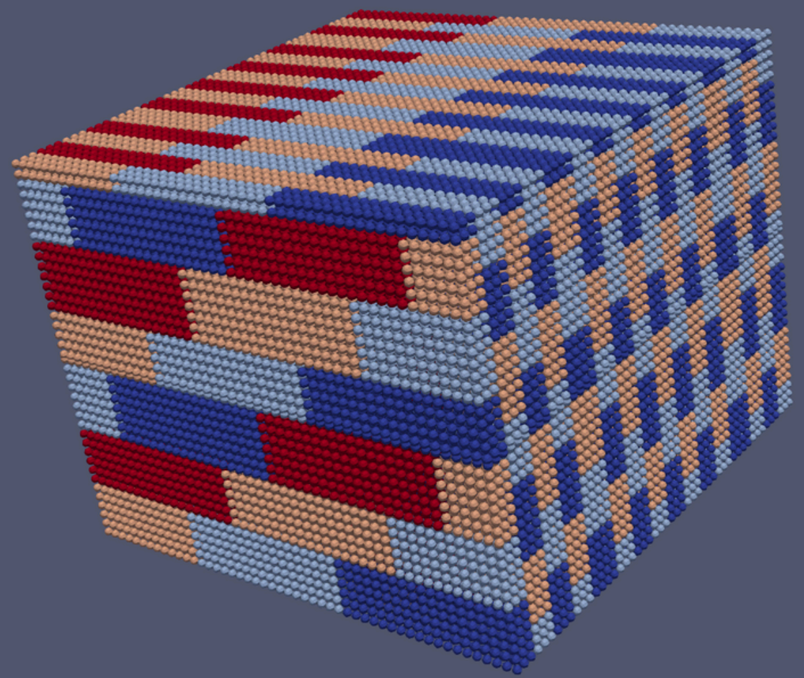
declarations of `#pragma omp`

Domain Coloring



Rayleigh-Taylor
3D cuboid

by color



```
cenario xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
 <header base_name="Rayleigh-Taylor instability"
         output_frequency="100"
         t_end="100" seed="1337"
         delta_t="0.0005"
         dimensions="3"
         vectorized="false"
         parallelized="true"
/>
<thermostat t_init="40" frequency="1000" brownian_motion="0"/>
<container>
    <linked_cells cutoff_radius="3.6">
        <domain_size x="60" y="60" z="60"/>
        <boundary_conditions>
            <boundary_condition type="periodic"/>
            <boundary_condition type="reflecting"/>
            <boundary_condition type="periodic"/>
            <boundary_condition type="periodic"/>
            <boundary_condition type="reflecting"/>
            <boundary_condition type="periodic"/>
        </boundary_conditions>
    </linked_cells>
    <!--        <vector/>-->
</container>
<forces>
    <lennard_jones>
        <gravity>-12.44</gravity>
        <particleTypes>
            <particleType id="0" sigma="1.2" epsilon="1" mass="1"/>
            <particleType id="1" sigma="1.1" epsilon="1" mass="2"/>
        </particleTypes>
        <particles>
            <cuboid spacing="1.2" particleTypeId="0">
                <coordinate x="0.6" y="0.6" z="0.6"/>
                <dimensions x="50" y="20" z="50"/>
                <velocity x="0" y="0" z="0"/>
            </cuboid>
            <cuboid spacing="1.2" particleTypeId="1">
                <coordinate x="0.6" y="24" z="0.6"/>
                <dimensions x="50" y="20" z="50"/>
                <velocity x="0" y="0" z="0"/>
            </cuboid>
```

TASK 3

Rayleigh-Taylor instability in 3D

Parallelization upfront

Implemented parallelization task beforehand, so the runtimes are manageable. Local runtime of around 42 seconds for contest file. The large experiment took about 8 hours on a laptop with 12 cores.

Simulation

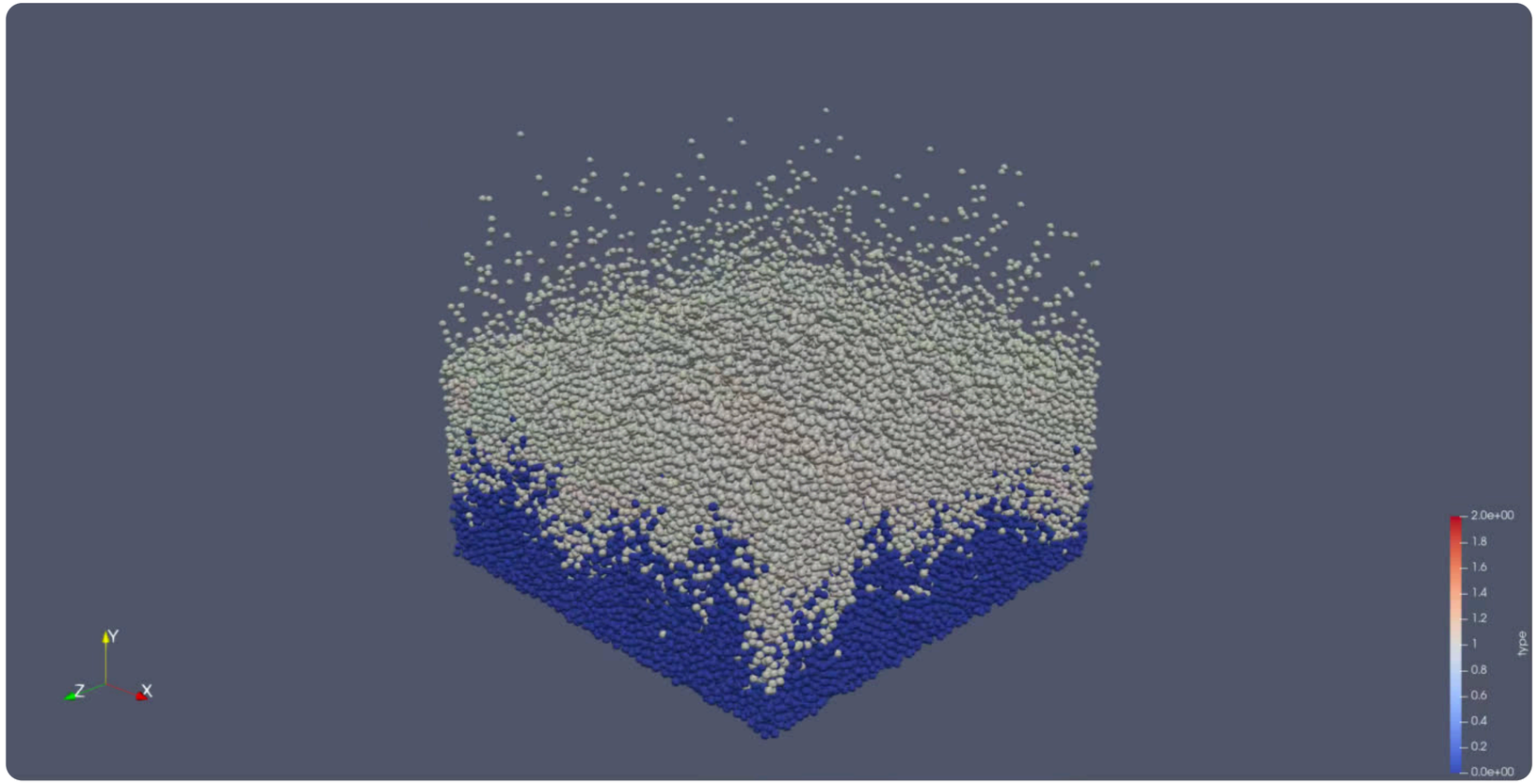
Input and force caclulation was already adapted for 3D simulations. Ran simulations according to worksheet from different angles and perspectives.

Local hardware: Ryzen 9 7950X3D 16K/32T, 64GB DDR5 RAM

SIMULATION 1

Colored by type (the denser fluid is white)

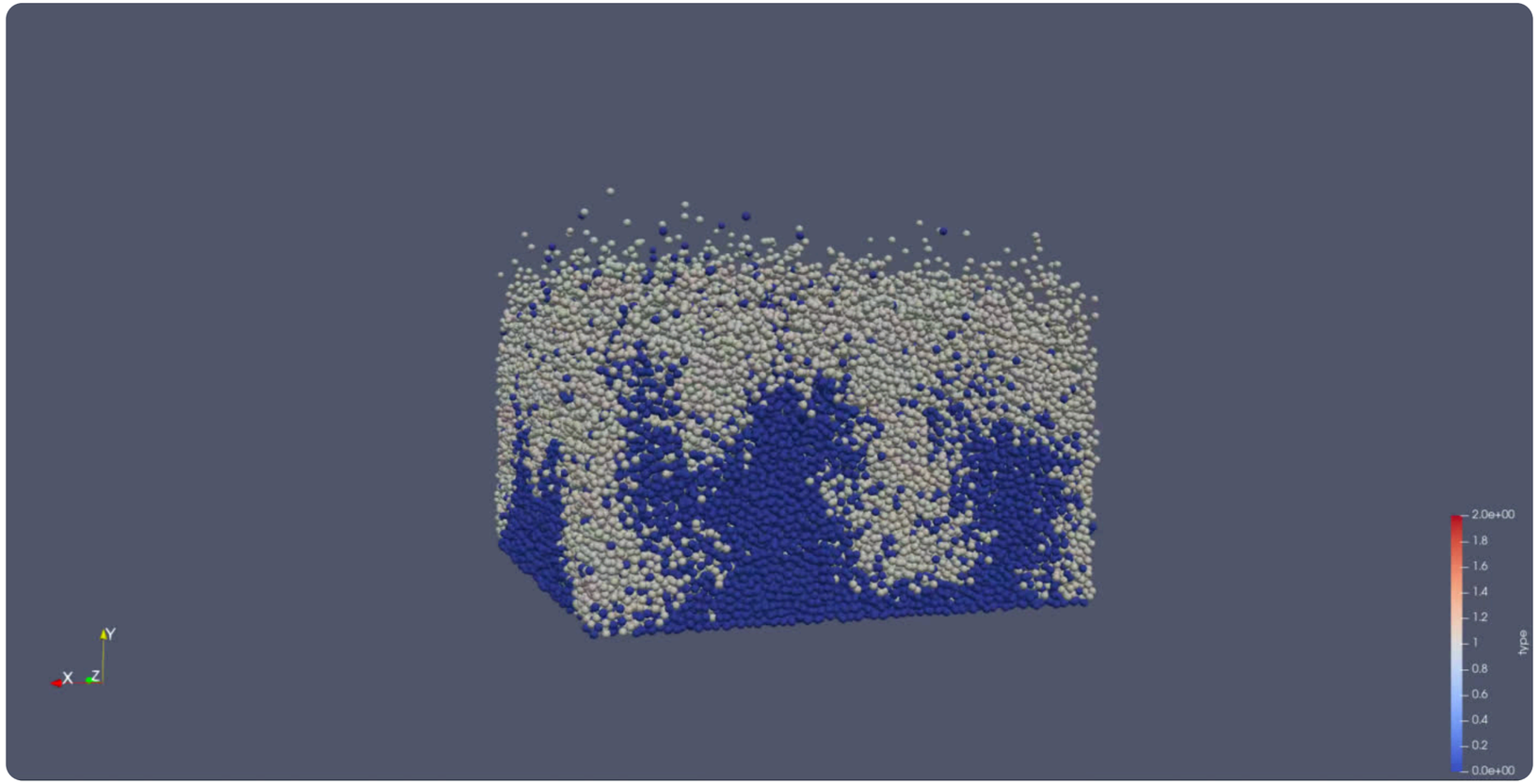
09



SIMULATION 1

Colored by type (the denser fluid is white)

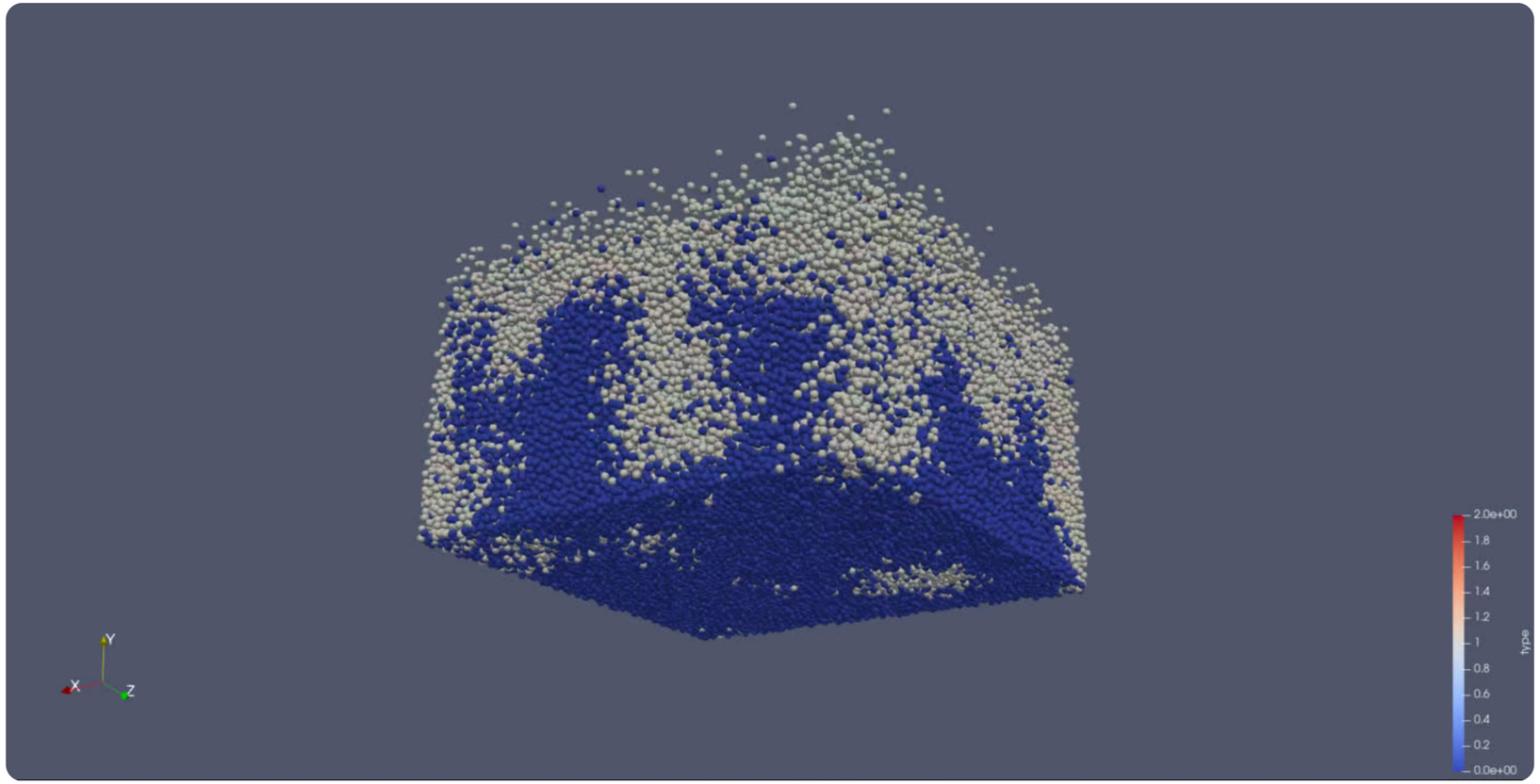
10



SIMULATION 1

Colored by type (the denser fluid is white)

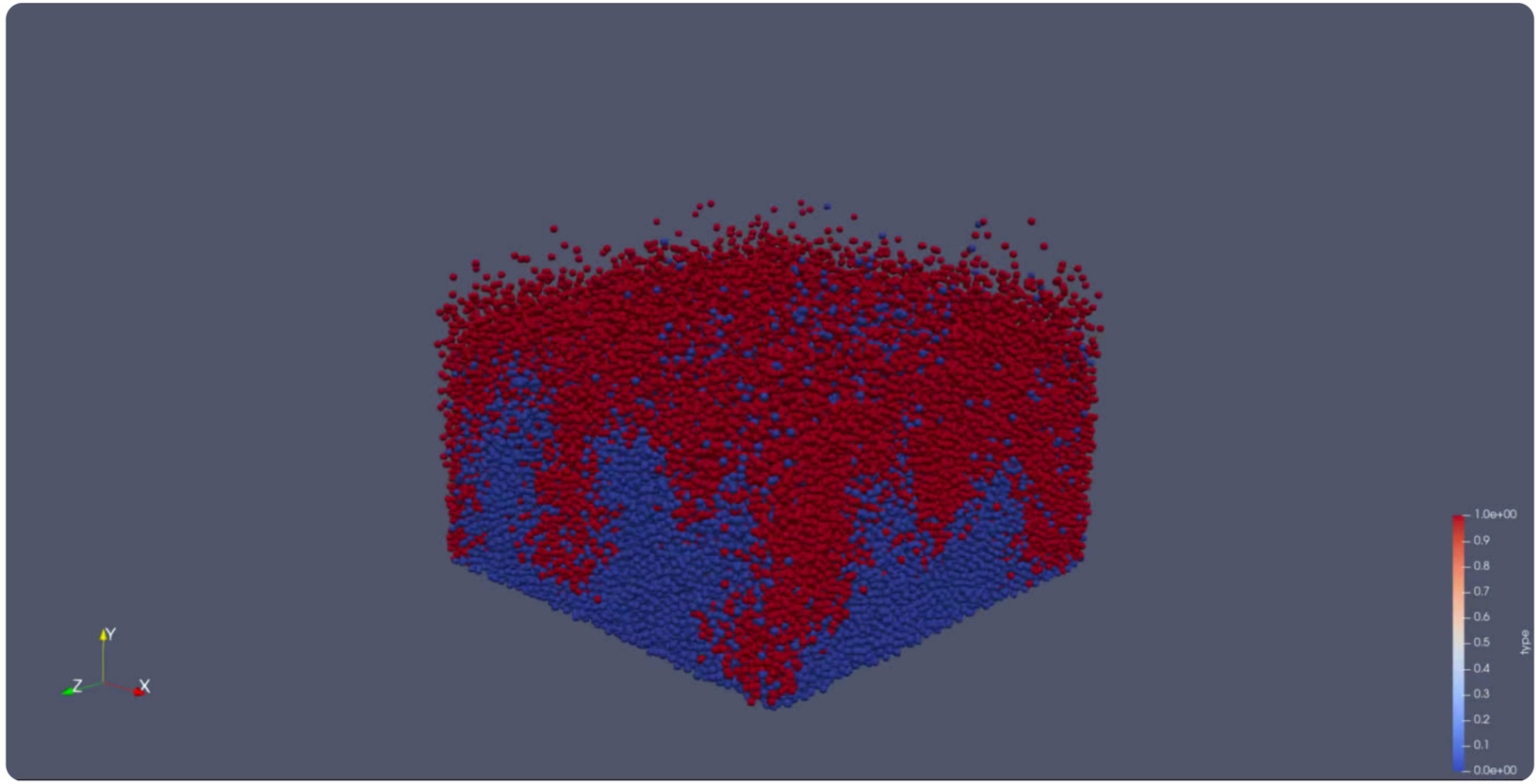
11



SIMULATION 1

Colored by type (the denser fluid is red)

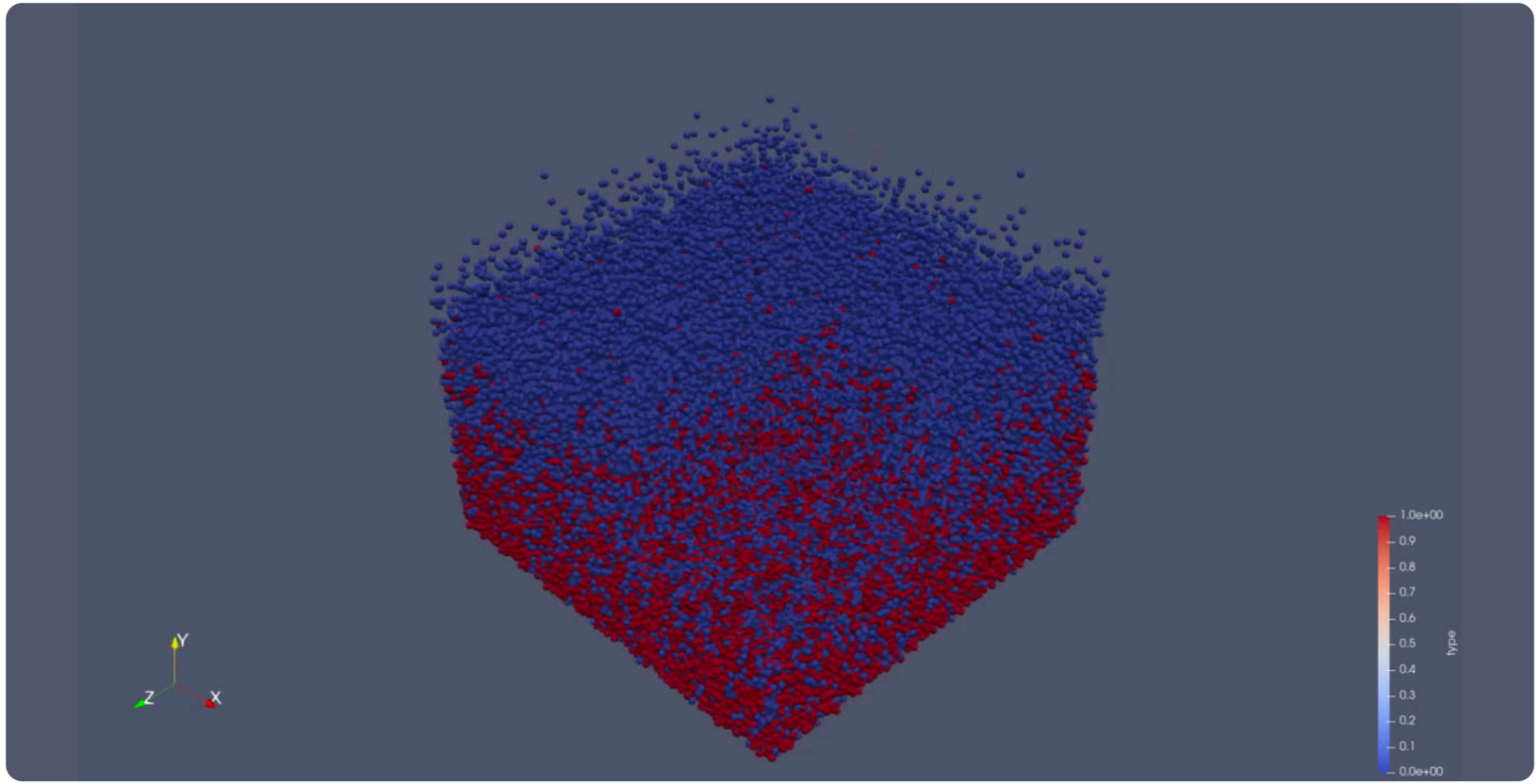
12



SIMULATION 1

Colored by type (the denser fluid is red)

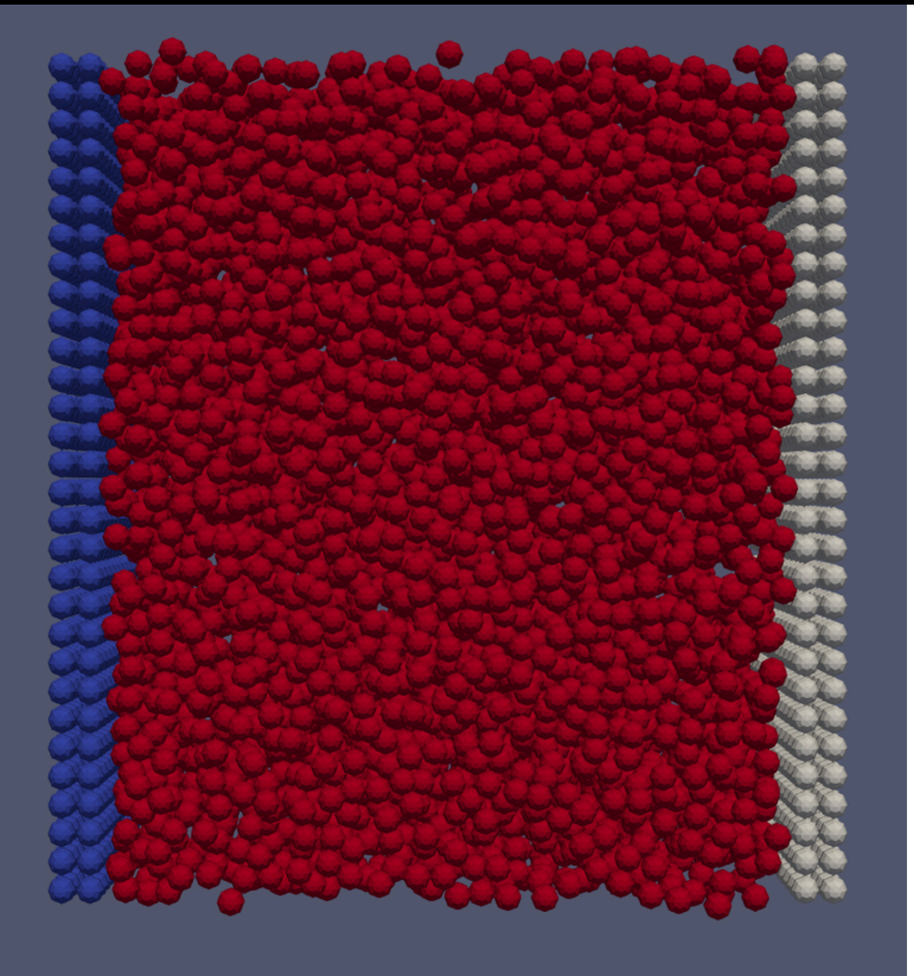
13



TASK 4 - OPTION A

Nano-scale flow simulation

Nanotube



Fixed walls

No position updates or velocity calculations for fixed molecules, implemented through an extra optional attribute in XML that is saved in config, specifying if a molcule is fixed

Different input parameters for sigma, epsilon etc. enable different types of materials, e.g. for the walls

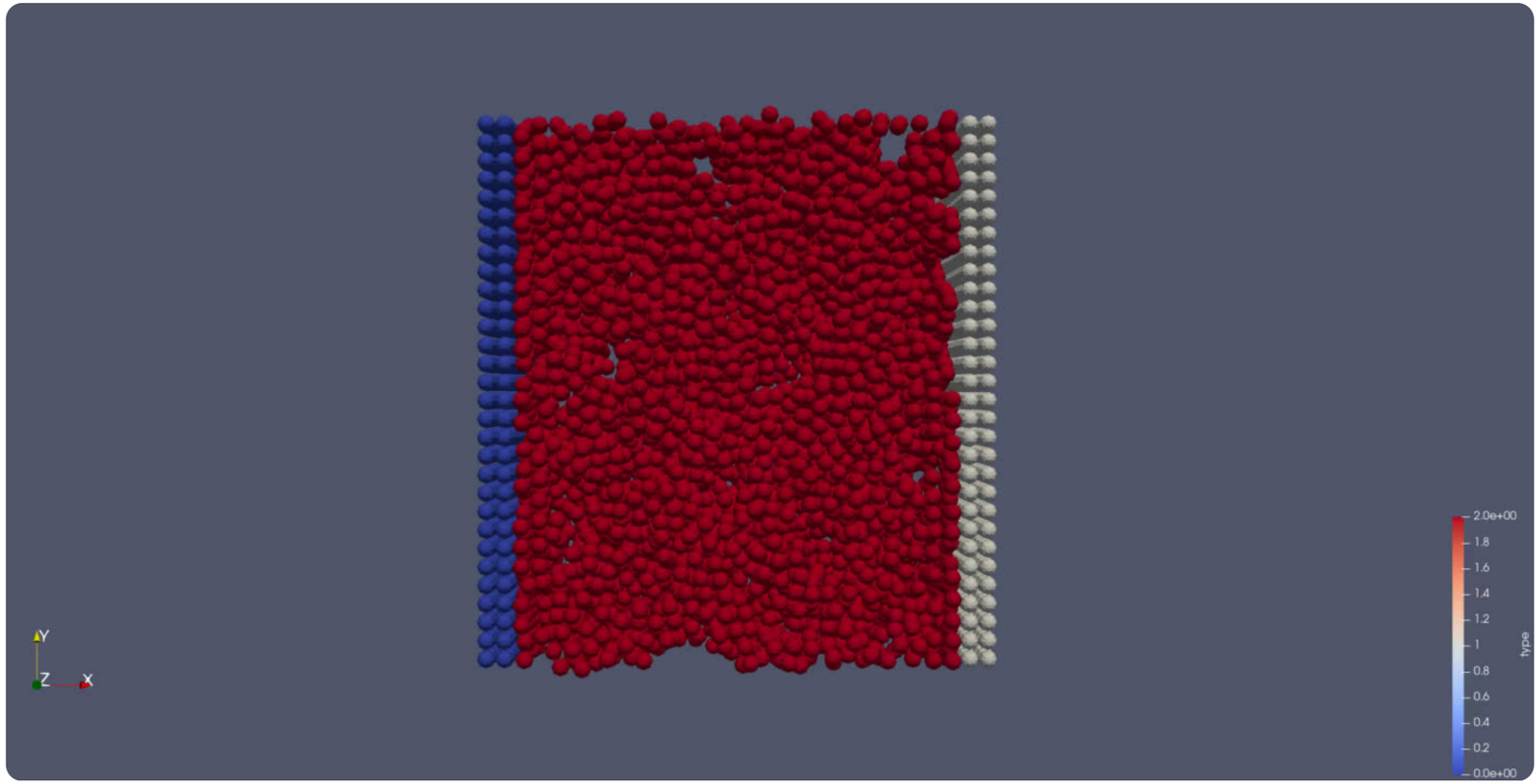
Performance

Local runtime for specifications on worksheet with output writing enabled but with t_end = 50 is 08:42:776 (mm:ss::ms)

SIMULATION 2

Colored by type (walls are blue and white, fluid is red)

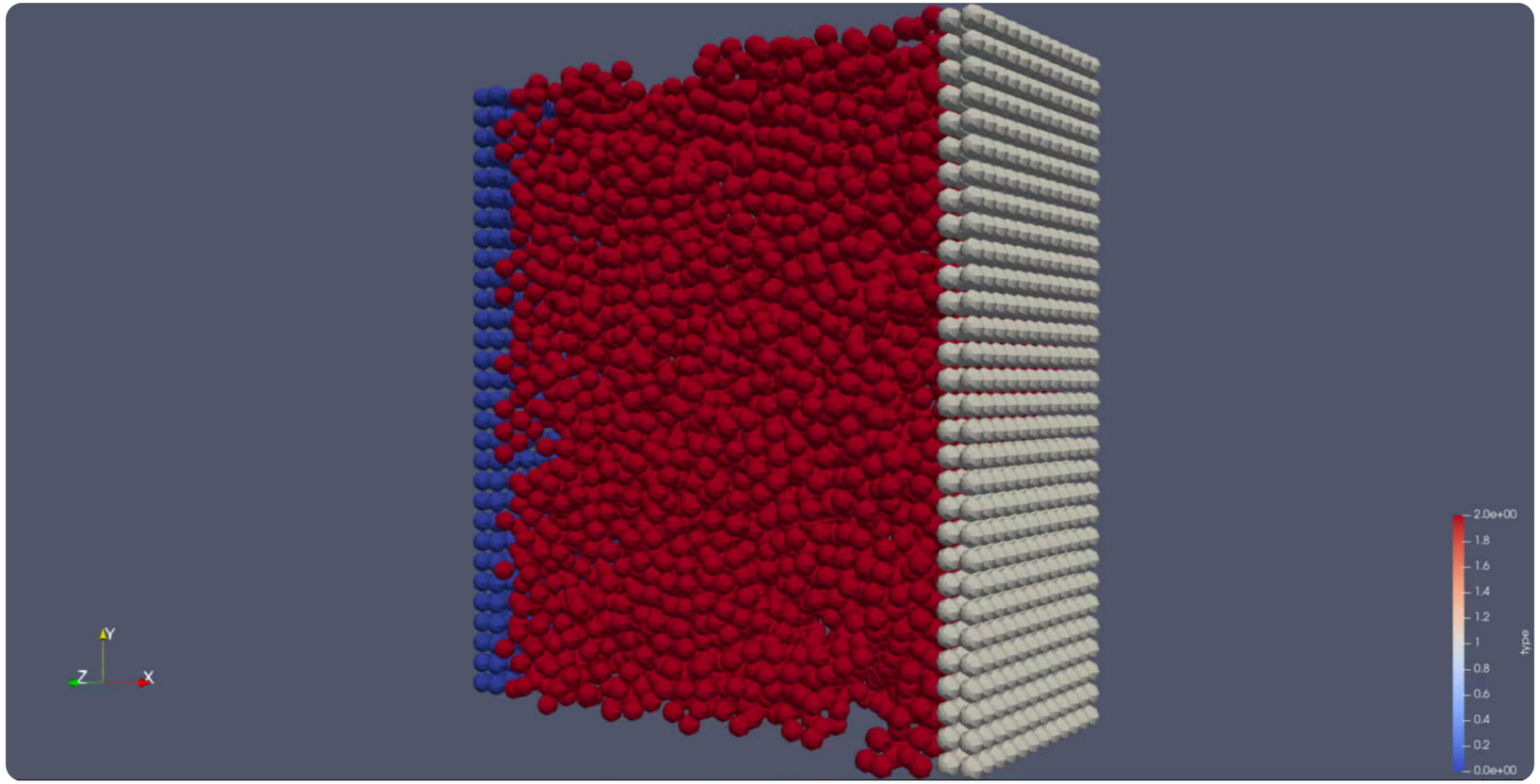
15



SIMULATION 2

Colored by type (walls are blue and white, fluid is red)

16

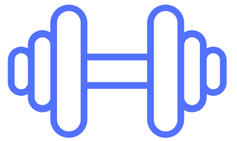


OTHER NANOTUBE SCENARIOS

Task 4 - Option A

High gravity

Large mass walls



Large epsilon fluid

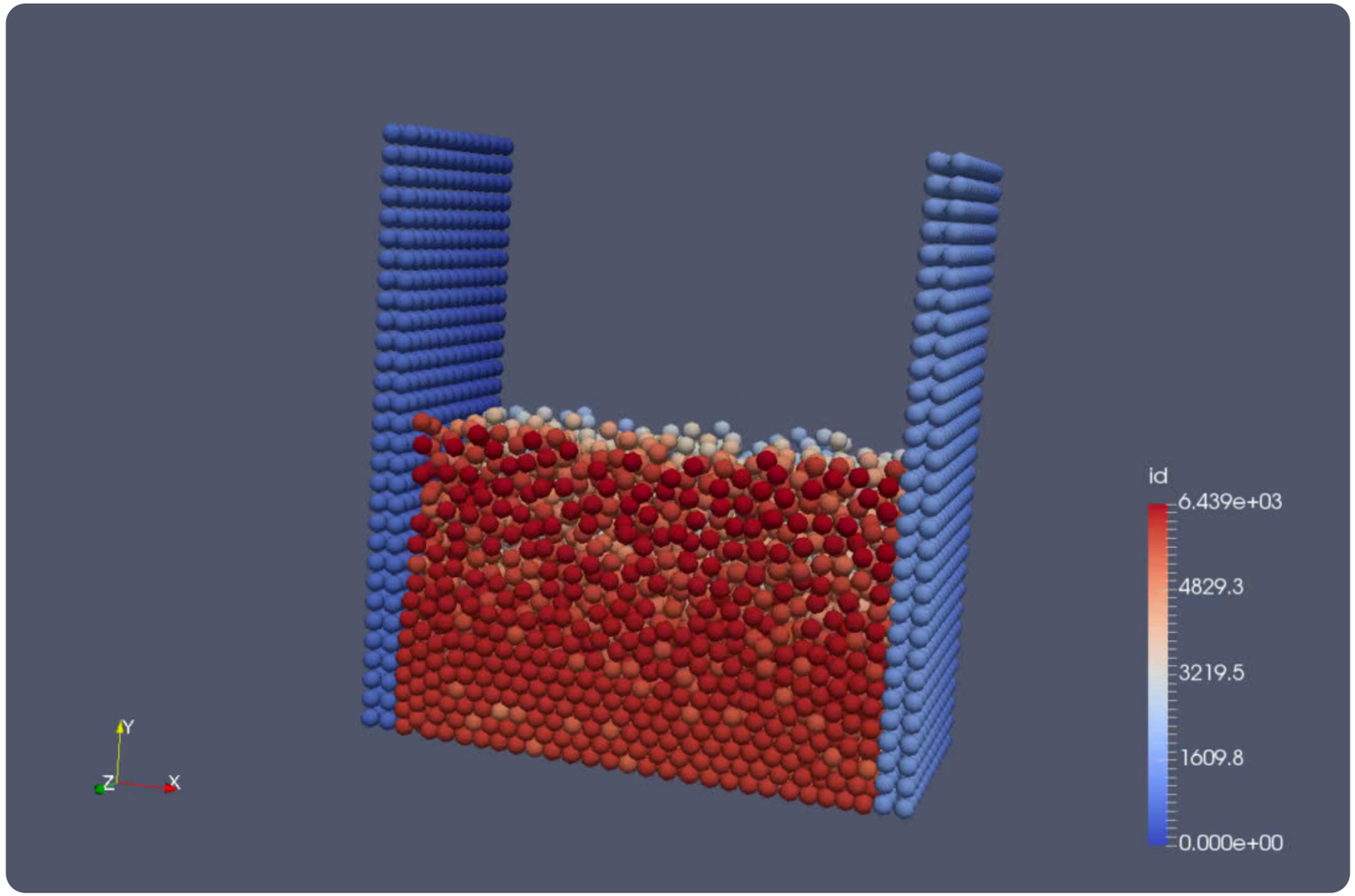
Large sigma fluid

SIMULATION 3

High gravity

Colored by id (walls are blue, fluid is red)

18

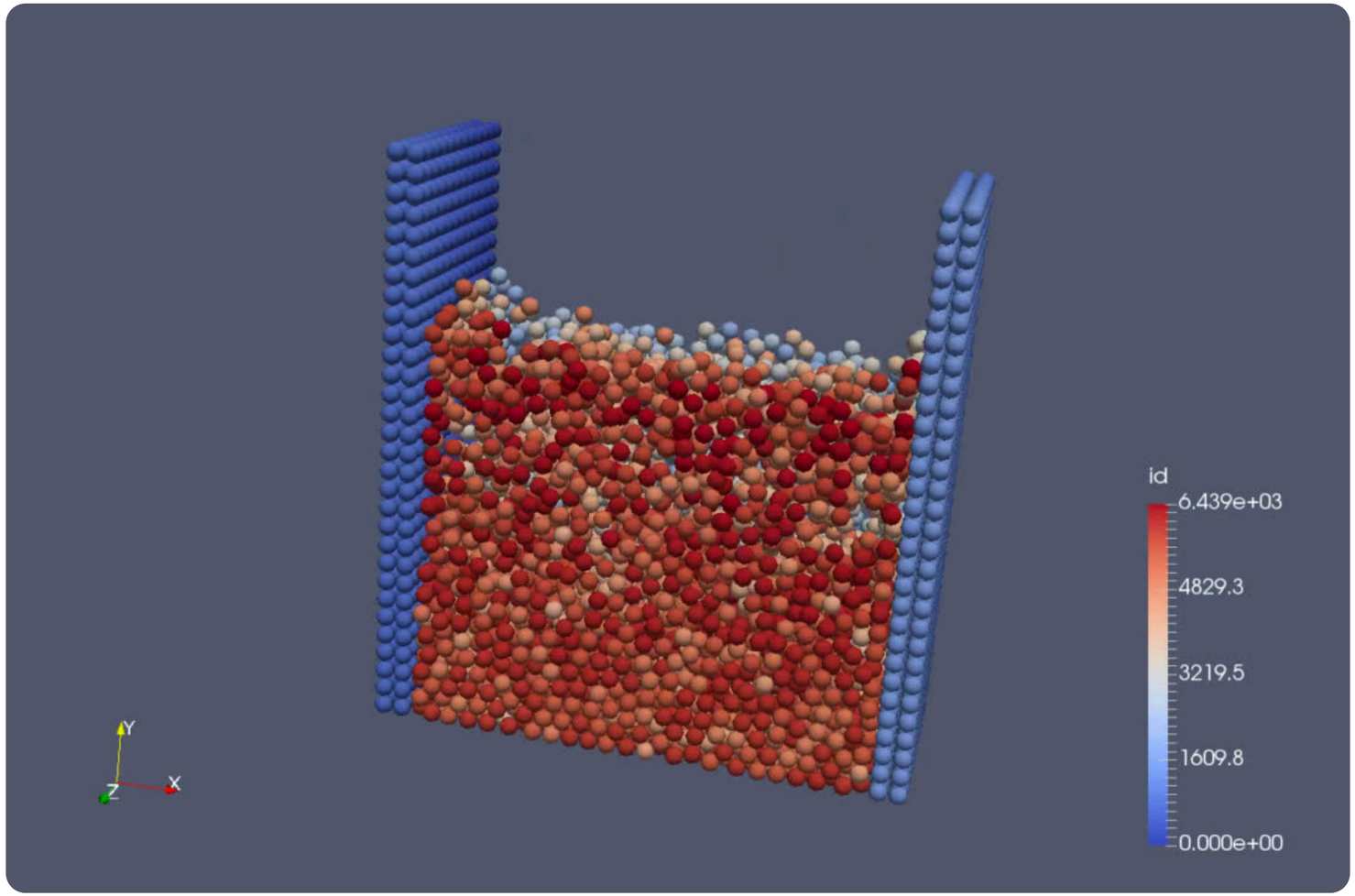


SIMULATION 4

Large mass walls

Colored by id (walls are blue, fluid is red)

19

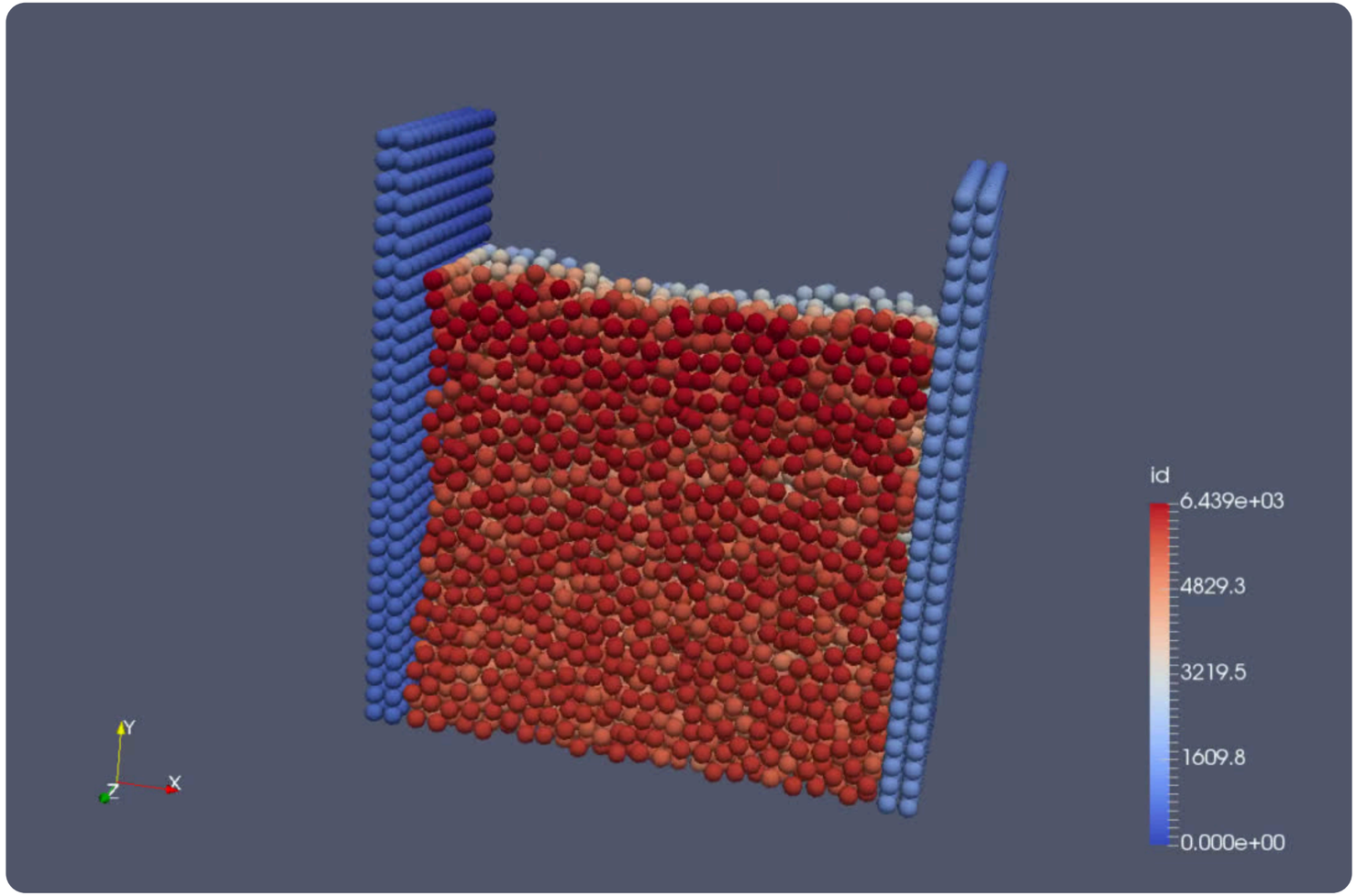


SIMULATION 5

Large epsilon fluid

Colored by id (walls are blue, fluid is red)

20

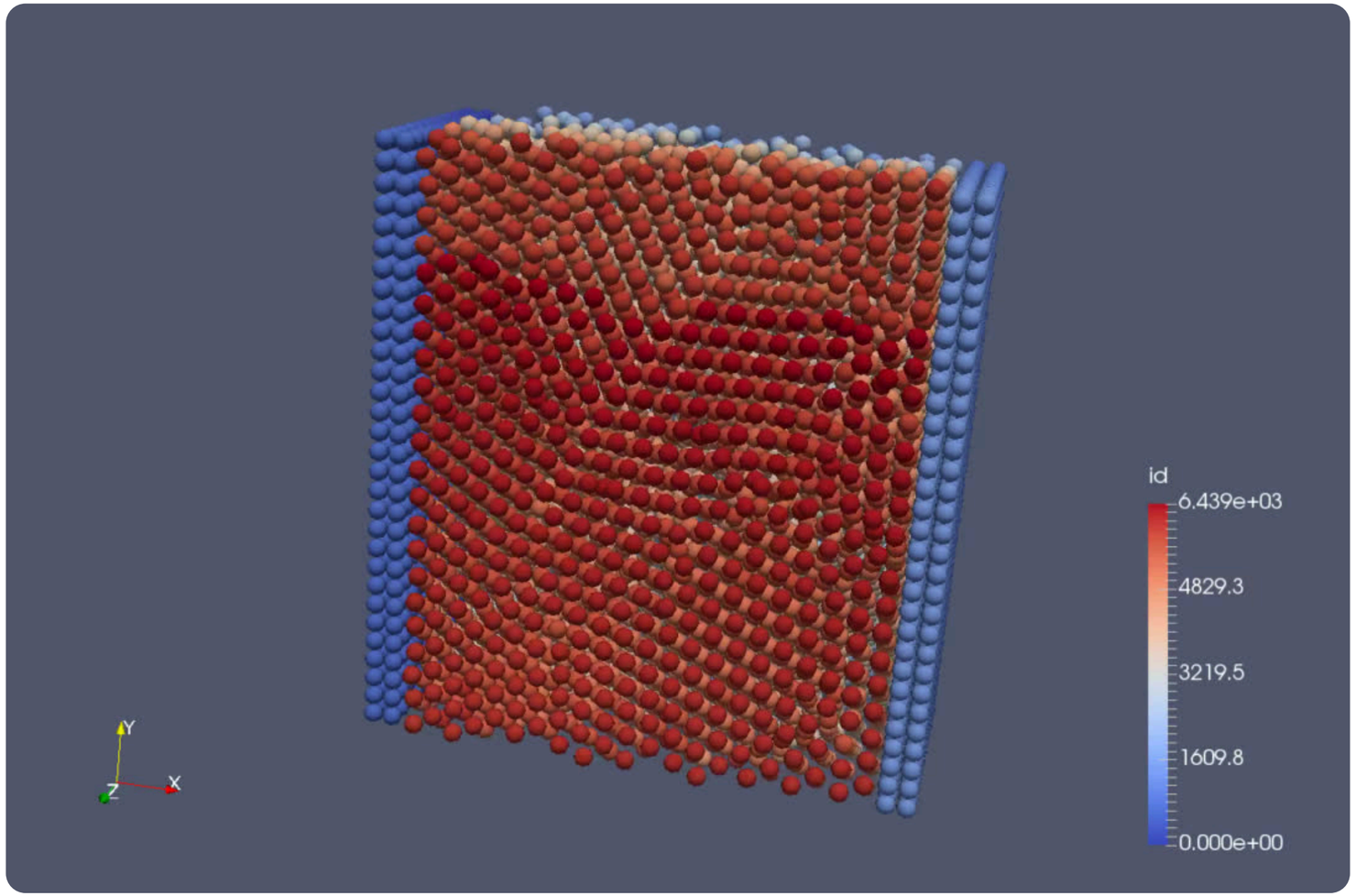


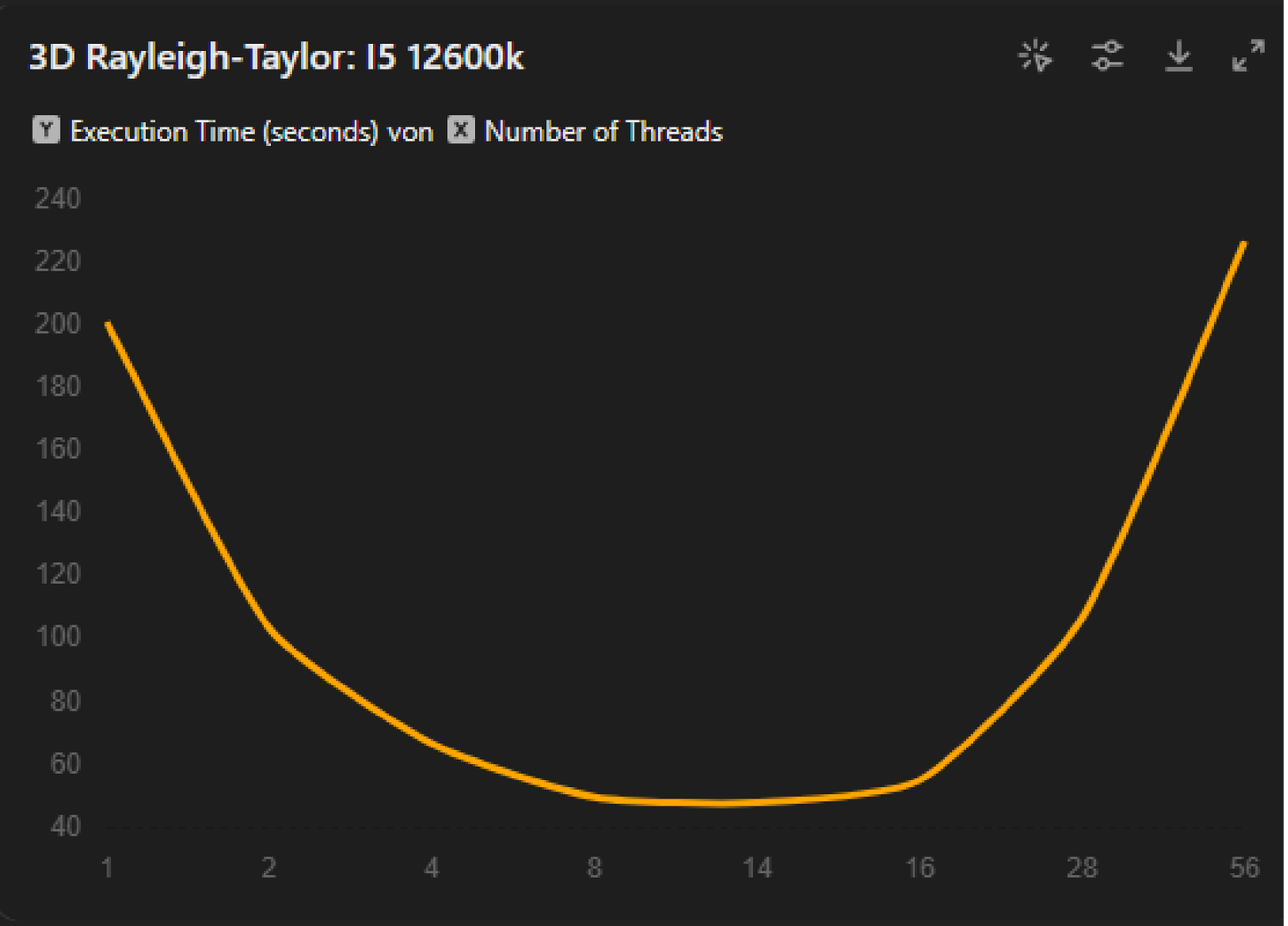
SIMULATION 6

Large sigma fluid

Colored by id (walls are blue, fluid is red)

21





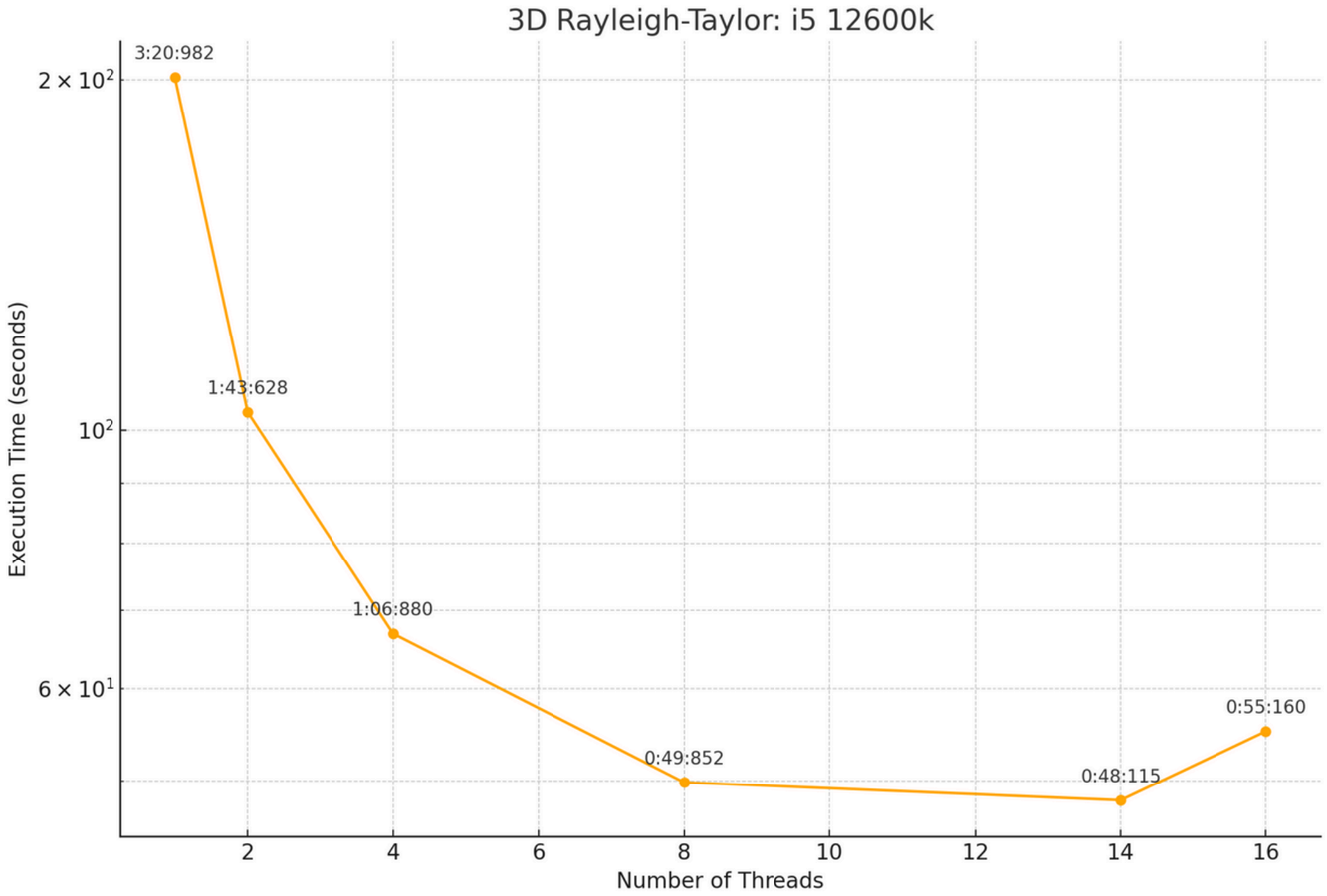
STRONG-SCALING PLOT

Calculated on local hardware for 1-16 “real” threads according to the used hardware

28 and 56 threads are not real threads but simulated as if they were

we did local computing due to problems on the cluster

Intel i5 12600K 10K/16T
32GB DDR4 RAM



STRONG-SCALING PLOT

Plot for the 1-16 actual threads

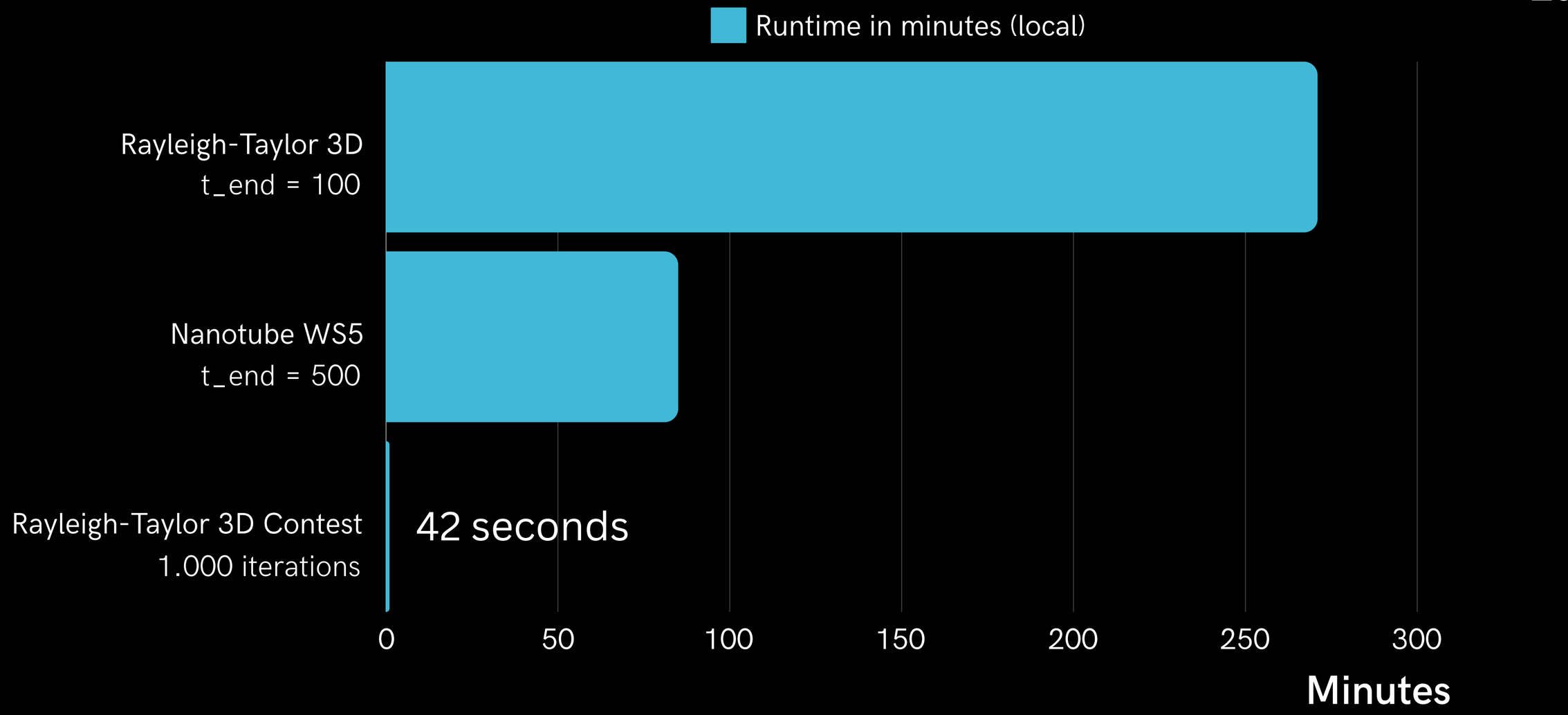
Intel i5 12600K 10K/16T
32GB DDR4 RAM

BENCHMARKS

Local Hardware: AMD Ryzen 9 7950X3D
16K/32T, 64GB RAM

Format
mm:ss:ms

	LOCAL MACHINE	CLUSTER
Rayleigh-Taylor Instability 3D	4-5 hours	did not run
Rayleigh-Taylor Instability 3D Contest	00:43:933	02:03:821
Nanoscale flow WS5	08:42:780	did not run



Local Hardware: AMD Ryzen 9 7950X3D 16K/32T, 64GB RAM

RAYLEIGH TAYLOR CONTEST 2

1000 iterations of the Taylor-Rayleigh instability simulation in 3D

02:03:821 (mm:ss:ms)
On linux cluster

00:43:76 (mm:ss:ms)
On local machine (Ryzen 9)

RAYLEIGH TAYLOR 3D WITH T-END 100

1000 iterations of the Taylor-Rayleigh instability simulation in 3D

approx. 4-5 hours
(unfortunately got interrupted)
On local machine (Ryzen 9)

08:41:23 (hh:mm:ss)
On local machine (Laptop 12K/24T)

CONTEST 2

RT 3D LONG

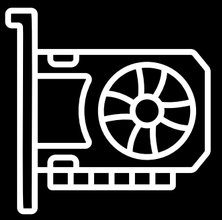
FUTURE IDEAS

LANGUAGE

Implementing the same / similar molecular simulator in Rust instead of C++ which has comparable or even faster performance and its own unique components

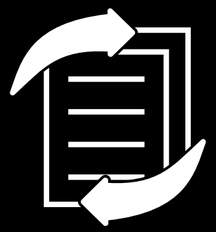
GPU PERFORMANCE

Optimizing the code for GPU computation to try out performance changes



REFACTORING

Rewriting the whole code based on the now known features it should support would yield better performance and streamlining



Scientific Computing (PSE) Molecular Dynamics
Group D

Johannes H. | Julius K. | Tim S.

THANK YOU FOR LISTENING