Molecular Simulation Group D

Worksheet 1

Generated by Doxygen 1.10.0

Chapter 1

MolSim2024 - Group D

1.1 Prerequisites

Developed and tested with these versions, other versions might work

- clang 17.0.6
- lldb 17.0.6
- cmake 3.27.7
- ninja 1.11.1

These environment variables are assumed to be always set. They are used to select which environment should be build. There purpose is simplifying the documentation only.

```
RELEASE_BUILD_DIR=build-release
DEBUG_BUILD_DIR=build-debug

BUILD_DIR=RELEASE_BUILD_DIR # or DEBUG_BUILD_DIR if debugging
```

Run this in the root folder of this repository

```
/usr/bin/cmake  -DCMAKE_BUILD_TYPE=Release \
                -DCMAKE_MAKE_PROGRAM=/usr/bin/ninja \
                -DCMAKE_CXX_COMPILER=/usr/bin/clang++ \
                -DCMAKE_EXPORT_COMPILE_COMMANDS=ON \
                -G Ninja \
                -S $(pwd) \
                -B $(pwd)/${BUILD_DIR}
```

‘-G 'Unix Makefile`` would generate make files if required

1.2 Building documentation

```
cd ${BUILD_DIR}
ninja doc_doxygen
```

1.3 Build and run

```
cd ${BUILD_DIR}
ninja
./MolSim
```

1.4 Clean

```
cd ${BUILD_DIR}
ninja clean
```

1.5 Static analysis

```
clang-tidy $files...$ -checks="cppcoreguidelines-*,modernize-*,performance-*,readability-*"  -p
      ./${BUILD_DIR}/compile_commands.json
```

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Class Documentation

4.1 FileReader Class Reference

`#include <FileReader.h>`

Public Member Functions

- FileReader ()
- virtual ~FileReader ()
- void readFile (std::vector< Particle > &particles, char *filename)

4.1.1 Constructor & Destructor Documentation

4.1.1.1 FileReader()

```
FileReader::FileReader ( )  [default]
```

4.1.1.2 ~FileReader()

```
FileReader::~FileReader ( )  [virtual], [default]
```

4.1.2 Member Function Documentation

4.1.2.1 readFile()

```
void FileReader::readFile (
            std::vector< Particle > & particles,
            char * filename )
```

The documentation for this class was generated from the following files:

- src/FileReader.h
- src/FileReader.cpp

4.2 Particle Struct Reference

`#include <Particle.h>`

Public Member Functions

- Particle (int type=0)
- Particle (const Particle &other)
- Particle (std::array< double, 3 > x_arg, std::array< double, 3 > v_arg, double m_arg, int type=0)
- ~Particle ()
- bool operator== (Particle &other) const
- std::string toString () const

Public Attributes

- std::array< double, 3 > x {}

  Position of the particle.
- std::array< double, 3 > v {}

  Velocity of the particle.
- std::array< double, 3 > f {}

  Force effective on this particle.
- std::array< double, 3 > old_f {}

  Force which was effective on this particle.
- double m {}

  Mass of this particle.
- int type

  Type of the particle.

4.2.1 Constructor & Destructor Documentation

4.2.1.1 Particle() [1/3]

```
Particle::Particle (
            int type = 0 )  [explicit]
```

4.2.1.2 Particle() [2/3]

```
Particle::Particle (
            const Particle & other )
```

4.2.1.3 Particle() [3/3]

```
Particle::Particle (
            std::array< double, 3 > x_arg,
            std::array< double, 3 > v_arg,
            double m_arg,
            int type = 0 )
```

4.2.1.4 ~Particle()

```
Particle::~Particle ( )
```

4.2.2 Member Function Documentation

4.2.2.1 operator==()

```
bool Particle::operator== (
            Particle & other ) const
```

4.2.2.2 toString()

```
std::string Particle::toString ( ) const
```

4.2.3 Member Data Documentation

4.2.3.1 f

```
std::array<double, 3> Particle::f {}
```

Force effective on this particle.

4.2.3.2 m

```
double Particle::m {}
```

Mass of this particle.

4.2.3.3 old_f

```
std::array<double, 3> Particle::old_f {}
```

Force which was effective on this particle.

4.2.3.4 type

```
int Particle::type
```

Type of the particle.

Use it for whatever you want (e.g. to separate molecules belonging to different bodies, matters, and so on)

4.2.3.5 v

```
std::array<double, 3> Particle::v {}
```

Velocity of the particle.

4.2.3.6 x

```
std::array<double, 3> Particle::x {}
```

Position of the particle.

The documentation for this struct was generated from the following files:

- src/Particle.h
- src/Particle.cpp

Chapter 5

File Documentation

5.1 README.md File Reference

5.2 src/FileReader.cpp File Reference

```
#include "FileReader.h"
#include <cstdlib>
#include <fstream>
#include <iostream>
#include <sstream>
```

5.3 src/FileReader.h File Reference

```
#include "Particle.h"
#include <vector>
```

Classes

- class FileReader

5.4 FileReader.h

Go to the documentation of this file.

```
00001 /*
00002  * FileReader.h
00003  *
00004  *  Created on: 23.02.2010
00005  *      Author: eckhardw
00006  */
00007
00008 #pragma once
00009
00010 #include "Particle.h"
00011
00012 #include <vector>
00013
00014 class FileReader {
00015
00016  public:
00017   FileReader();
00018   virtual ~FileReader();
00019
00020   void readFile(std::vector<Particle> &particles, char *filename);
00021 };
```

5.5 src/MolSim.cpp File Reference

```
#include "FileReader.h"
#include "outputWriter/VTKWriter.h"
#include "outputWriter/XYZWriter.h"
#include "utils/ArrayUtils.h"
#include <iostream>
```

Functions

- void calculateF ()

  calculates the force for all particles
- void calculateX ()

  calculates the position for all particles
- void calculateV ()

  calculates the position for all particles
- void plotParticles (int iteration)

  plotting function that takes an integer value and has no return value
- int main (int argc, char ∗argsv[ ])

  main function for execution

Variables

- constexpr double start_time = 0

  initialisation of the start time of the simulation with 0
- constexpr double end_time = 1000

  initialisation of the end time of the simulation with 1000 according to the worksheet task 4.1
- constexpr double delta_t = 0.014

  time delta of 0.014 according to the worksheet task 4.1
- std::vector< Particle > particles

5.5.1 Function Documentation

5.5.1.1 calculateF()

```
void calculateF ( )
```

calculates the force for all particles

5.5.1.2 calculateV()

```
void calculateV ( )
```

calculates the position for all particles

5.5.1.3 calculateX()

```
void calculateX ( )
```

calculates the position for all particles

5.5.1.4 main()

```
int main (
             int argc,
             char * argsv[] )
```

main function for execution

Parameters

argc	an integer argument, standard for main function
argsv	a char array argument, standard for main function

Returns

the return code

5.5.1.5 plotParticles()

```
void plotParticles (
             int iteration )
```

plotting function that takes an integer value and has no return value

Parameters

iteration	an integer argument that sets the number of iterations

5.5.2 Variable Documentation

5.5.2.1 delta_t

```
constexpr double delta_t = 0.014  [constexpr]
```

time delta of 0.014 according to the worksheet task 4.1

5.5.2.2 end_time

```
constexpr double end_time = 1000  [constexpr]
```

initialisation of the end time of the simulation with 1000 according to the worksheet task 4.1

5.5.2.3 particles

```
std::vector<Particle> particles
```

5.5.2.4 start_time

```
constexpr double start_time = 0  [constexpr]
```

initialisation of the start time of the simulation with 0

5.6 src/Particle.cpp File Reference

```
#include "Particle.h"
#include "utils/ArrayUtils.h"
#include <iostream>
```

Functions

- std::ostream & operator<< (std::ostream &stream, Particle &p)

5.6.1 Function Documentation

5.6.1.1 operator<<()

```
std::ostream & operator<< (
            std::ostream & stream,
            Particle & p )
```

5.7 src/Particle.h File Reference

```
#include <array>
#include <string>
```

Classes

- struct Particle

Functions

- std::ostream & operator<< (std::ostream &stream, Particle &p)

5.7.1 Function Documentation

5.7.1.1 operator<<()

```
std::ostream & operator<< (
            std::ostream & stream,
            Particle & p )
```

5.8 Particle.h

Go to the documentation of this file.

```
00001 /*
00002  * Particle.h
00003  *
00004  *  Created on: 23.02.2010
00005  *      Author: eckhardw
00006  */
00007
00008 #pragma once
00009
00010 #include <array>
00011 #include <string>
00012
00013 struct Particle {
00017   std::array<double, 3> x{};
00018
00022   std::array<double, 3> v{};
00023
00027   std::array<double, 3> f{};
00028
00032   std::array<double, 3> old_f{};
00033
00037   double m{};
00038
00043   int type;
00044
00045  public:
00046   explicit Particle(int type = 0);
00047
00048   Particle(const Particle& other);
00049
00050
00051   Particle(
00052       // for visualization, we need always 3 coordinates
00053       // -> in case of 2d, we use only the first and the second
00054       std::array<double, 3> x_arg, std::array<double, 3> v_arg, double m_arg,
00055       int type = 0);
00056
00057   ~Particle() ;
00058
00059   bool operator==(Particle &other) const;
00060
00061   std::string toString() const;
00062 };
00063
00064 std::ostream &operator«(std::ostream &stream, Particle &p);
```