



DPD: A Java implementation

Jo Hoffmann

EPCC, University of Edinburgh
 University of Edinburgh
 email jxh@dcs.ed.ac.uk

The aim of this project was to write a Java code for a DPD simulation.

DPD physics

DPD stands for Dissipative Particle Dynamics and is similar to Molecular Dynamics (MD) in terms of algorithm development.

The state of a DPD simulation is entirely defined by the position of all particles and their momenta. The simulation has two steps

- force calculation
- velocity and position update

The choice of a language

In general languages of choice for HPC applications are Fortran, C or C++. For many people Java is not yet a language of choice for HPC. Features that Java boasts are:

- Object Oriented language which allows flexible code reuse and modularisation
- Portability: Byte-code can run on any platform that has a Java Virtual Machine
- Network facilities like Remote Method Invocation and Jini provide facilities for distributed computing.
- GUI support is build in

A language with so many features can not be ignored. But still many people believe that the performance losses of Java codes compared to their Fortran or C equivalents are reason enough. But benchmarking results [1] show that the performance gap is closing down especially with JIT compilers becoming better and better.

The Class-model

The simulation has 3 main components

- Particle
- CompoundParticle
- Space

The Particle classes deal with single particles. They are the 'atoms' of the simulation because they are used on their own and within the CompoundParticles. Force calculation can only take place between 2 Particles. The only final class is Monomer. In case other particles that have different interactions need to be implemented a new class extending DPDPparticle can be added.

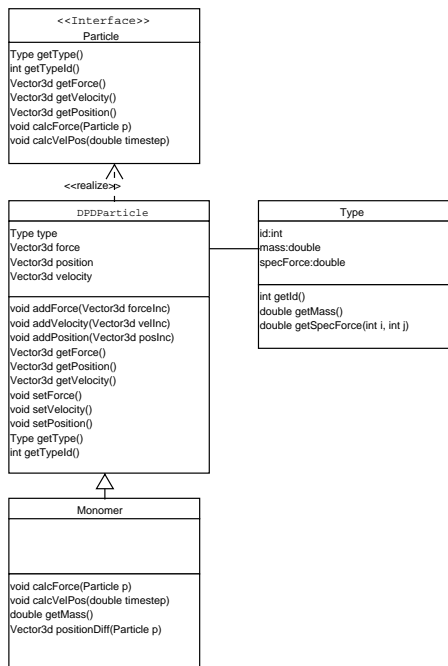


Figure 1 : The Particle hierarchy

A CompoundParticle is a particle structure. It is responsible for updating the velocity and position of its individual Particles while maintaining the constraints (like distance between Particles for Dimers) under which the structure is assembled.

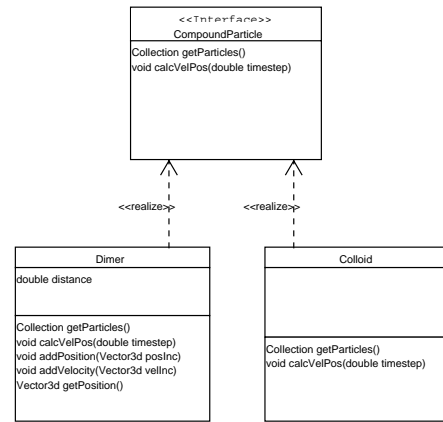


Figure 2 : The CompoundParticle hierarchy

The Space classes are the simulation managers. They setup and execute the simulation according to parameters provided in a file. In order to implement different boundary conditions a new class extending DPDSpace can be added.

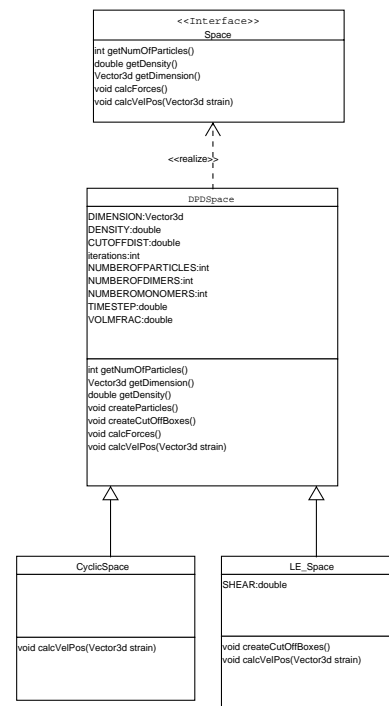


Figure 3 : The Space hierarchy

Conclusion

The result of 10 weeks of work is a working serial code with documentation. Remaining work will be the development of a GUI and the parallelisation of the code.

Acknowledgements

I would like to thank my two supervisors Lorna Smith from EPCC and Alexander Wagner from the Physics Department for the help and advice they gave me throughout this project. A full report of the project can be found at <http://www.epcc.ed.ac.uk/ssp/>.

References

- 1 J.M. Bull, L.A. Smith, L. Pottage and R. Freeman, Benchmarking Java against C and Fortran for Scientific Applications, In Proceedings of ACM Java Grande/ISCOPE Conference, June 2001