Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Computer Science Department

Prof. Dr. Peter Arbenz

Parallelization of the discrete element method

Proposal for a master thesis / industry project

Introduction

The *discrete element method* (DEM) is a popular approach for computing the motion and effect of a large number of small particles. The DEM is becoming widely accepted as an effective method for addressing engineering problems in granular and discontinuous materials. It implements the explicit multi-body dynamics with a large numbers of bodies.
The DEM is used, e.g., for the simulation of fracture and fragmentation of concrete when a chisel is applied, see Fig. 1, or when a screw is driven in the material.

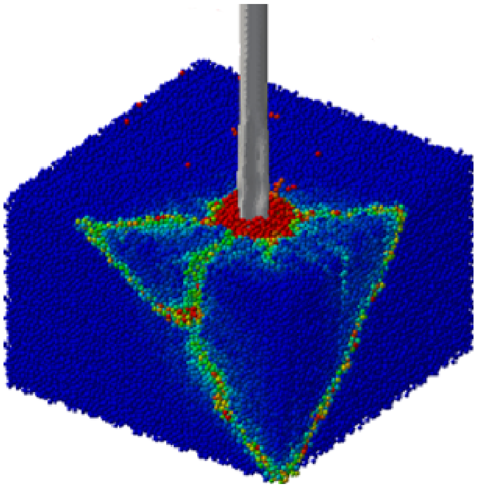

Figure 1: Fracturing of a block of concrete by a chisel tool.

Scope of work

The starting point of this work is an implementation of the discrete element method in Yade [6]. Yade is an extensible open-source framework for discrete numerical models, focused on the Discrete Element Method. The computation parts are written in C++ using flexible object model, allowing the independent implementation of new alogrithms and interfaces. Python is used for rapid and concise scene construction, simulation control, postprocessing and debugging[1]. The

[1] http://yade-dem.org/doc/

available code is able to simulate fracture and fragmentation of concrete. It employs very many ($10^5 - 10^9$) rigid bodies to model the geometry of the concrete structure and its mechanical properties. The available code moves these particles based on Newton's law of motion by means of external and inter-particle forces described by contact laws, e.g. a cohesive bond law modeling the specific behavior of concrete. An explicit time stepping method is used. The contact detection of n particles has complexity $n \log(n)$ (instead of n^2).
The structure of the algorithm is to a large extent identical with known algorithms for particle-based simulations [3]. We therefore envision that we can use known open codes for particle simulation like GROMACS[2] for molecular dynamics, IPPL[3] for for the simulation of particle movements in accelerator physics or astronomy, or LAMMPS which is a classical molecular dynamics code for Large-scale Atomic/Molecular Massively Parallel Simulations[4].
The use of GROMACS in molecular simulations is discussed by Páll, Abraham *et al.* [1, 5]. IPPL (Independent Parallel Particle Layer) is an object-oriented framework for particle and field based applications in computational science requiring high-performance parallel computers [2]. In LAMMPS some models treat particles as finite-size spheres, as opposed to point particles. (This is comparable to the treatment of particles in Yade.) This means they have an angular velocity and torque can be imparted to them to cause them to rotate, see [4].
The work will be split in two parts.

1. The shorter first part (1-2 months) deals with the suitability of these particle frameworks for the planned Yade applications. A framework for the highly parallel handling of big numbers of particles is selected.
2. In the larger second part of the work (3-4 months) a prototype for the interaction of Yade with the chosen particle framework is to be implemented.

   A test application is provided to develop this prototype. A realistic application is provided to investigate if it is suitible for larger scale applications.

Requirements

- Student in computational science or related fields.
- Good knowledge in numerical mathematics.
- Good command of C++ and literacy in Python.

Deliverables

- The work is to be documented in a short and concise thesis (LaTeX, PDF). It must be written such that it is intelligible to a fellow-student.
- The code should be written as clean as possible. It must be properly documented.
- At the end of the thesis, the work is to be presented in a 30 minutes' talk.

Contact

- Prof. Dr. Peter Arbenz, Computer Science, ETH Zurich, arbenz@inf.ethz.ch
- Dr. Michael Pertl, Coorporate Research & Technology, Hilti AG, Michael.Pertl@hilti.com

[2] http://www.gromacs.org/
[3] http://amas.web.psi.ch/tools/IPPL/
[4] http://lammps.sandia.gov/

References

[1] M. J. Abraham, Murtola T., Schulz R., S. Páll, J. C. Smith, B. Hess, and E. Lindahl. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1-2:19–25, 2015.

[2] A. Adelmann. The IPPL (Independent Parallel Particle Layer) framework. Technical Report PSI-PR-09-05, Paul Scherrer Institute, 2009. Release of July 3, 2012 available at http://amas.web.psi.ch/docs/ippl-doc/ippl_user_guide.pdf.

[3] R. W. Hockney and J. W. Eastwood. *Computer Simulation using Particles*. Institute of Physics Publishing, Bristol, 1988.

[4] LAMMPS Documentation, April 2016. http://lammps.sandia.gov/doc/Manual.html.

[5] S. Páll, M. J. Abraham, C. Kutzner, B. Hess, and E. Lindahl. Tackling exascale software challenges in molecular dynamics simulations with GROMACS. In S. Markidis and E. Laure, editors, *Solving Software Challenges for Exascale: International Conference on Exascale Applications and Software, EASC 2014*, pages 3–27. Springer, 2015.

[6] V. Šmilauer et al. *Yade Documentation, 2nd ed.* The Yade Project, November 2015. Available from http://yade-dem.org/doc/.

April 20, 2016. P. Arbenz