

Data structures and algorithms for contact detection in numerical simulation of discrete particle systems

V. Ogarko and S. Luding

Multi Scale Mechanics/TS, Faculty of Engineering Technology, University of Twente,
7500 AE, Enschede, The Netherlands.

v.ogarko@utwente.nl; s.luding@utwente.nl; <http://www.msm.ctw.utwente.nl>

January, 2010.

1 Introduction

Collision detection is a basic computational problem arising in systems consisting of many objects, particles or atoms. It is fundamental to many applications, including computer games, physically based simulations and others.

The Discrete Element Method (DEM) is a very enticing simulation technique for computing the motion of a large number of particles in granular media [1]. Since a large number of particles imply a long computation time, for realistic systems, high performance computing is mandatory. The performance of computations relies on several factors, which include the physical model and the contact detection algorithm used.

The simple straight-forward approach works by checking each pair of particles for collision. This requires $O(N^2)$ collision checks for N particles and is not practicable. More efficient collision detection methods use a two-phase approach to reduce the computational costs [2].

2 Data structures

There are four groups of the broad phase collision detection methods based on different data structures: spatial sorting methods [3], grids (or cell-based methods) [4,5], trees [6] and Delaunay triangulation [7,8]. The choice of algorithm depends on several factors (i.e. system size, volume fraction, homogeneity, boundary conditions, polydispersity, dynamics) and should be taken accordingly to the considered problem. It is not enough to know about the average theoretical complexity and memory consumption of the algorithm, but it is necessary to analyze how an algorithm works with existing data. For example, while for the simulations of monodisperse homogeneous systems the Linked Cells method [5] performs as $O(N)$ and has near $O(N)$ memory consumption for the dense systems, it becomes less efficient with increasing polydispersity [9] and consumes a lot of memory for the dilute and non-homogeneous systems (if no special techniques of storing grid cells are used). In this work we briefly analyze three methods, which are related to different data structures: hierarchical grid, Delaunay triangulation and Octree. Our interest is to simulate dilute, intermediate and dense particles systems with mono-, bi- and polydisperse size distributions, therefore, the algorithm should perform well for all cases, so that it can be applied for a wide range of cutting edge physical problems: segregation, clustering, jamming, fragmentation, two-phase flow and others.

2.1 Hierarchical grid

We propose a hierarchical hashed grid (HHGrid) algorithm [10] based on hierarchical spatial hashing, for details see Refs. [2,11]. The algorithm is adjusted for DEM simulations, by taking into account that all particles are in motion. We improved the collision scheme so that less particle pairs have to be checked and the number of checks is independent of hash collisions. Furthermore, we analyzed how to select good parameters according to the particle size distribution. Finally, several DEM simulations with realistic physical systems are performed as test-cases [10].

The algorithm is robust, consumes $O(N)$ memory for all type of systems, considered by us, performs as $O(N)$ and most important, works well for systems with wide polydispersity. It can be easily used with periodic boundaries, be simply parallelized and can accommodate unbounded systems due to the hashing approach. Adding and deleting particles from the system is fast, therefore parts of huge industrial dynamic systems like conveyer belts, silos, or pneumatic conveying systems and especially systems with fragmentation can be simulated too.

2.2 Weighted Delaunay triangulation

Delaunay triangulation (DT) is another attractive data structure. First, it can be used not just for contact detection, but also for several purposes, i.e. strain calculation [12], space meshing and others. Second, it has an advantage that selecting parameters according to the radii distribution is not necessary, because the particle size is included directly in the calculation. Third, it consumes $O(N)$, namely, $(44-80) \cdot N$ bytes of memory [13] (depends on storage data structure). In contrast, cell-based methods require to store a d -dimensional grid, whose size depends on the size of the system, so that additional techniques have to be used (i.e. hashing) to decrease memory consumption.

Simple Delaunay triangulation may miss contacts for polydispersity higher than ~ 2.41 in 2D and 3D (for a proof see Ref. [14]). Weighted Delaunay triangulation (WDT) can detect contacts whenever a condition on the maximum overlap for the particles $1 \leq i \leq N$ is satisfied [8], with no overhead against simple Delaunay:

$$\forall (i, j) \in \{1, \dots, N\}^2, i \neq j \Rightarrow \|x_i - x_j\|^2 > R_i^2 + R_j^2, \quad (1)$$

where x_i is the position and R_i is the radius of particle i . We can assume that (1) is always true since interparticle overlap is usually very small in realistic DEM simulations.

Despite the fact that DT has higher theoretical complexity than HHGrid, we found that a flipping algorithm for maintaining triangulation [15] increases performance dramatically in 2D. Unfortunately in 3D flipping can get “stuck”, and when this happens, triangulation should be rebuilt from scratch. The “stuck”-ing frequency highly depends on the input data [16]. Although the DT algorithm is not robust, because triangle orientation and “InCircle” tests are used, we found that the algorithm did not fail for our data and hope that the effect of nonrobustness can be neglected for DEM simulations. Furthermore DT cannot be easily parallelized and handling periodic boundary conditions is not as simple as for cell-based methods. Adding and especially deleting particles is expensive, so for handling dynamic systems WDT can be inefficient.

2.3 Octree

Tree data structures seem to be very similar to hierarchical grids, but there are many small details, that potentially cause problems, which set them apart, i.e. the existence of a root incurs computational and memory costs; access to neighbors is difficult; unbounded systems can not be easily accommodated; cell size (subcubes are used to generate the data structure) at different level cannot be easily adjusted to the particle size distribution. Within this study we use the pointer-based Octree algorithm, as described in Ref. [11], to compare performance with the two previous methods HHGrid and WDT.

3 Results

Here we present a performance comparison between different contact detection methods for molecular dynamics simulations of homogeneous systems with walls in 2D and 3D with soft spherical elastic particles. For the purpose of comparison for all methods a common C++ framework was developed. The CGAL external library was used for the construction of the WDT.

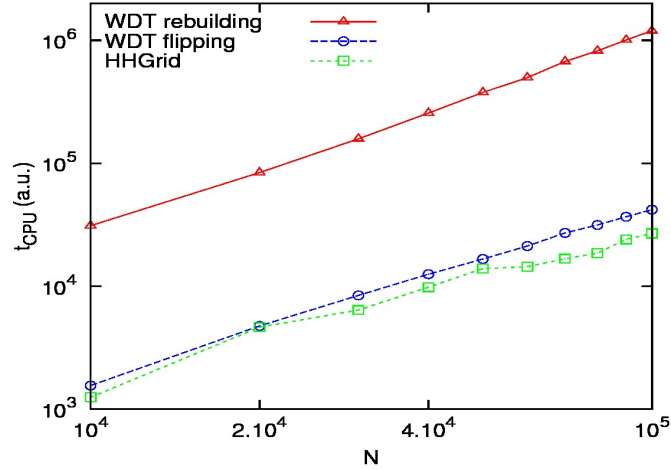


Fig.1. Log-log plot: CPU time comparison in 2D, uniform polydispersity 1:10, volume fraction 0.55 for $N \leq 5 \cdot 10^4$, 0.35 for $N > 5 \cdot 10^4$.

As can be seen from Fig.1, the use of a flipping algorithm increases performance of WDT dramatically as compared to rebuilding the triangulation from scratch every time step. Note, that CGAL implies an overhead due to its interval arithmetic, but it is not more than 2-3 times, while the performance gain is ~ 12 times. The results (3D) in Fig.2 show that HHGrid is fastest, while WDT (without flipping) is slowest, both with linear scaling, as in 2D (for $N \geq 4 \cdot 10^4$). Octree is slower than HHGrid, partly because no particular optimization was performed.

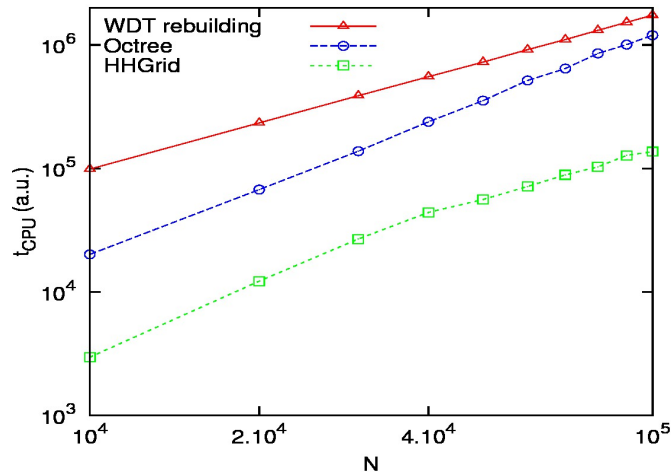


Fig.2. Log-log plot: CPU time comparison in 3D, uniform polydispersity 1:10, volume fraction 0.6.

4 Summary

Hierarchical grid contact detection is fast, flexible and memory efficient. It can be applied for a wide range of physical problems (see Ref. [10]). Its performance is governed by the choice of parameters, as described in Ref. [10]. WDT is not as fast but with use of the flipping algorithm it becomes very competitive – at least in 2D. WDT adjusts to the particles size distribution by itself and can be used for several purposes simultaneously. Whereas theoretically, the flipping procedure in 2D converges in at most $O(N^2)$ steps, we found that in practice it is $\sim O(N)$. The WDT flipping algorithm in 3D has to be examined in more detail. We have not found advantages of the Octree data structure for contact detection in DEM simulations. A more detailed comparison, based on realistic physical systems with different N , high polydispersity and various volume fractions can be found in Ref. [14].

Acknowledgement

This research is supported by the Dutch Technology Foundation STW, which is the applied science division of NWO, and the Technology Programme of the Ministry of Economic Affairs (10120).

References

- [1] S. Luding, Introduction to Discrete Element Methods: Basics of Contact Force Models and how to perform the Micro-Macro Transition to Continuum Theory, EJECE (Alert Course Lecture, Aussois), 785-826 (2008).
- [2] B. Mirtich, Efficient algorithms for two-phase collision detection. TR-97-23, Mitsubishi Electric Research Laboratory (1997).
- [3] D. Baraff, "Dynamic Simulation of Non-penetrating Rigid Bodies," Ph.D. thesis, Technical Report 92-1275, Computer Science Dept., Cornell University, 1992.
- [4] M.P. Allen and D.J. Tildesley. Computer Simulation of Liquids (Clarendon, Oxford, 1990)
- [5] A. Munjiza, K.R.F. Andrews, 1998. "NBS contact detection algorithm for bodies of similar size", Int. J. of Numer. Methods and Eng., Vol. 43 pp.131-49.
- [6] J. Barnes, P. Hut. A hierarchical $O(N \log N)$ force calculation algorithm. Nature, 324:446-449, 1986.
- [7] J.-A. Ferrez, D. Müller, Th.M. Liebling, Dynamic Triangulations for Granular Media Simulations, in Statistical Physics and Spatial Statistics, The Art of Analyzing and Modeling Spatial Structures and Pattern Formation, K.R. Mecke and D. Stoyan (Eds.), Lecture Notes in Physics, Springer, 2000, ISBN 3-540-67750-X.
- [8] L. Pournin, On the behavior of spherical and non-spherical grain assemblies, its modeling and numerical simulation, Ph.D. thesis, École Polytechnique Fédérale de Lausanne (2005).
- [9] B. Muth, M.-K. Müller, P. Eberhard and S. Luding. Collision Detection and Administration Methods for Many Particles with Different Sizes. In: B. Wills (ed.) Proceedings of MEI. Brisbane, Australia; 2007. p. 1–18.
- [10] V. Ogarko, S. Luding. Fast, robust and memory efficient collision detection algorithm for granular media simulations with different sizes. (in preparation).
- [11] C. Ericson, Real-time collision detection. 2005. Morgan Kaufmann, Elsevier, Los Altos, CA, Amsterdam.
- [12] O. Durán, N.P. Kruyt, S. Luding, 2010. Analysis of three-dimensional micro-mechanical strain formulations for granular materials: evaluation of accuracy. International Journal of Solids and Structures 47 251-260.
- [13] A.V. Skvortsov, Delaunay triangulation and its applications (Izd. Tomsk. Gos. Univ., Tomsk, 2002).
- [14] V. Ogarko, S. Luding. Techniques for contact detection in granular matter simulations. (in preparation).
- [15] C. Lawson, Transforming triangulations. Discr. Math. 1972. N. 3. P. 365–372.
- [16] L. Guibas, D. Russel, An Empirical Comparison of Techniques for Updating Delaunay Triangulations, ACM Symp. on Comp. Geometry, pp. 170-179, 2004.