A fast multilevel algorithm for contact detection of arbitrarily polydisperse objects

V. Ogarko, S. Luding

Multi Scale Mechanics (MSM), CTW, UTwente,
PO Box 217, 7500 AE Enschede, Netherlands;
v.ogarko@utwente.nl, s.luding@utwente.nl

Abstract

We develop an efficient algorithm for contact detection among many arbitrarily sized objects. Objects are allocated to cells based on their location and size within a nested hierarchical cell space. The choice of optimal cell sizes and the number of hierarchies for best performance is not trivial in most cases. To overcome this challenge, a novel analytical method to determine the optimal hierarchical cell space for a given object size distribution is presented. With this, a decision can be made between using the classical linked-cell method and the contact detection algorithm presented. For polydisperse systems with size ratios up to 50, we achieved 220 times speed-up compared to the classical Linked-Cell method. For larger size ratios, even better speed-up is expected. The complexity of the algorithm is linear with the number of objects when the optimal hierarchical cell space is chosen. So that the problem of contact detection in polydisperse systems essentially is solved.

Keywords: contact detection, discrete element, polydisperse, different sizes, particle size distribution

1. Introduction

Collision detection is a basic computational problem arising in computer simulations of systems consisting of many discrete objects such as particles or atoms. The particle based modeling methods like the Discrete Element Method (DEM) [1] or Smoothed Particle Hydrodynamics [2] play an important role for physics-based simulations in various fields. The performance of the computation relies on several factors, which include the physical model, on the one hand, and the contact detection algorithm used, on the other. The collision detection of short-range pairwise interactions between particles is usually one of the most time-consuming tasks in calculations [3].

The most commonly used method for contact detection of nearly monosized particles with short-ranged forces is the Linked-Cell method [4, 5]. Due to its simplicity and high performance, it has been utilized since the beginning of particle simulations, and is easily implemented in parallel codes [6, 7].

Nevertheless, the Linked-Cell method is unable to efficiently deal with particles of greatly varying sizes [8]. This can effectively be addressed by the use of methods based on *hierarchical grids* [8, 9, 10, 11, 12, 13, 14]. Most of these methods can be assigned to two groups. In the first, the contacts between particles from different hierarchy levels are detected in the coarse grid [9, 10, 11, 13, 14], while in the second group the detection is done in the fine grid [8]; our method corresponds to the latter group. An extensive review of various approaches to contact detection is given in Ref. [15]. The performance difference between them is studied in Refs. [16, 17, 18].

Even though various other methods using hierarchical grid structures have been suggested, we improve upon these methods by (i) reducing the range of the contact search, and (ii) using two freely adjustable parameters: the number of hierarchy levels and the cells' size at each level. An analytical method to select these parameters is also developed. This method (for choosing parameters) is designed to improve the performance of the algorithm and can be used for an arbitrary polydisperse particle size distribution. We confirm our theoretical predictions by means of DEM simulations of homogeneous and isotropic systems of elastic spherical particles, even though the algorithm is not limited to these ideal cases.

The paper is organized as follows. Section 2 outlines the algorithm. We then present the method how to choose the optimal parameters in Section 3. Section 4 presents the performance results of the numerical simulations. Finally, the results are summarized and discussed, with some conclusions in Section 5.

2. Algorithm

The present algorithm is designed to determine all the pairs in a set of N spherical particles in a d-dimensional Euclidean space that overlap. Every particle is characterized by the position of its centre $\vec{x}_p$ and its radius r_p. For differently-sized spheres r_{min} and r_{max} denote the minimum and the maximum particle radius, respectively, and $\omega = r_{max}/r_{min}$ is the extreme size ratio (size distribution functions used are explained in section 4).

The algorithm is made up of two phases. In the first "mapping phase" all the particles are mapped into a hierarchical grid space (subsection 2.1). In the second "contact detection phase" (subsection 2.2) for every particle in the system the potential contact partners are determined, and the geometrical intersection tests with them are made.

2.1. Mapping phase

The d-dimensional *hierarchical grid* is a set of L regular grids with different cell sizes. Every regular grid is associated with a hierarchy level $h \in [1, L]$, where L is the integer number of hierarchy levels. Each level h has a different cell size $s_h \in \mathbb{R}$, where the cells are d-dimensional cubes. Grids are ordered with increasing cell size so that $h = 1$ corresponds to the grid with smallest cell size, i.e. $s_h < s_{h+1}$. For a given number of levels and cell sizes, the hierarchical grid cells are defined by the following spatial mapping, M, of points $\vec{x} \in \mathbb{R}^d$ to a cell at specified level h:

$$M : \ (\vec{x}, h) \mapsto \vec{c} = (\lfloor x_1/s_h \rfloor, ..., \lfloor x_d/s_h \rfloor, h), \qquad (1)$$

where $\lfloor x \rfloor$ denotes the floor function [1]. The first d components of a $(d + 1)$-dimensional vector $\vec{c}$ represent cell indices (integers), and the last one is the associated level of hierarchy. The latter is limited whereas the former are not.

It must be noted that the cell size of each level can be set independently, in contrast to contact detection methods which use a tree structure for partitioning the domain [10, 18, 19], where the cell sizes are taken as double the size of the previous lower level of hierarchy, hence $s_{h+1} = 2s_h$. The flexibility of independent s_h allows one to select the optimal cell sizes, according to the particle size distribution, to improve the performance of the simulations. How to do this is explained in Section 3.

Using the mapping M, every particle p can be mapped to its cell:

$$\vec{c}_p = M(\vec{x}_p, h(p)), \qquad (2)$$

where $h(p)$ is the *level of insertion* to which particle p is mapped to. The level of insertion $h(p)$ is the lowest level where the cell is big enough to contain the particle p:

$$h(p) = \left\{ \min_{1 \le h \le L} h : s_h \ge 2r_p \right\}. \qquad (3)$$

In this way the diameter of particle p is smaller or equal to the cell size in the level of insertion and therefore the classical Linked-Cell method [5] can be used to detect the contacts among particles within the same level of hierarchy.

Figure 1 illustrates a 2-dimensional two-level grid for the special case of a bi-disperse system with $r_{min} = 3/2$, size ratio $\omega = 8/3$, and cell sizes $s_1 = 3$, and $s_2 = 8$. Since the system contains particles of only two different sizes, two hierarchy levels are sufficient here.

2.2. Contact detection phase

The contact detection is split into two steps, and the search is done by looping over all particles p and performing the first and second steps consecutively for each p. The first step is the contact search at the level of insertion of p, $h(p)$, using the classical Linked-Cell method [5]. The search is done in the cell where p is mapped to, i.e., $\vec{c}_p$, and in its neighbour (surrounding) cells. Only half of the surrounding cells are searched, to avoid testing the same particle pair twice.

[1] the largest integer not greater than x

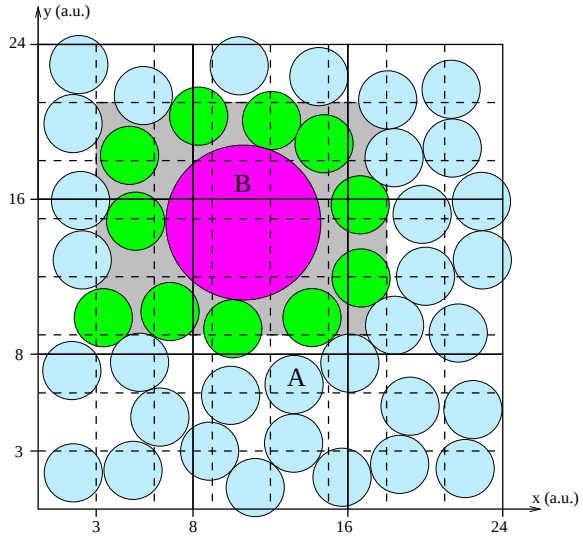

Figure 1: A 2-dimensional two-level grid for the special case of a bi-disperse system with cell sizes $s_1 = 2r_{min} = 3$ (a.u.), and $s_2 = 2r_{max} = 8$ (a.u.). The first level grid is plotted with dashed lines while the second level is plotted with solid lines. The radius of the particle B is $r_B = 4$ (a.u.) and its position is $\vec{x}_B = (10.3, 14.4)$. Therefore, according to Eqs. (2) and (3), particle B is mapped to the second level to the cell $\vec{c}_B = (1, 1, 2)$. Correspondingly, particle A is mapped to the cell $\vec{c}_A = (4, 2, 1)$. The cells where the cross-level search for particle B has to be performed from (1,3,1) to (5,6,1) are marked in grey, and the small particles which are located in those cells are dark (green). Note, that in the method of Iwai *et al* [8] the search region starts at cell $(1, 2, 1)$, i.e., one more layer of cells (which also includes particle A).

The second step is the *cross-level search*. For a given particle p, one searches for potential contacts only at levels h lower than the level of insertion: $1 \le h < h(p)$. This implies that the particle p will be checked only against the smaller ones, thus avoiding double checks for the same pair of particles. The cross-level search for particle p (located at $h(p)$) with level h is detailed here:

1. Define the cells $\vec{c}^{\,start}$ and $\vec{c}^{\,end}$ at level h as

$$\vec{c}^{\,start} := M(\vec{x}_c^{\,-}, h), \text{ and } \vec{c}^{\,end} := M(\vec{x}_c^{\,+}, h), \qquad (4)$$

   where a search box (cube in 3D) is defined by $\vec{x}_c^{\,\pm} = \vec{x}_p \pm \alpha \sum_{i=1}^{d} \mathbf{e}_i$, with $\alpha = r_p + 0.5 s_h$ and $\mathbf{e}_i$ is the standard basis for $\mathbb{R}^d$. Any particle q from level h, i.e., $h(q) = h$, with center $\vec{x}_q$ outside this box can not be in contact with p, since the diameter of the largest particle at this level can not exceed s_h. In Fig. 2 the grey colored cells correspond to the cells $\vec{c}^{\,start}$ (left bottom) and $\vec{c}^{\,end}$ (right top) for particle B from the situation shown in Fig. 1.
2. The search for potential contacts is performed in every cell $\vec{c} = (c_1, ..., c_d, h)$ for which

$$c_i^{start} \le c_i \le c_i^{end} \text{ for all } i \in [1, d], \text{ and } c_{d+1} = h < h(p), \qquad (5)$$

   where c_i denotes the i-th component of vector $\vec{c}$. In other words, each particle which was mapped to one of these neighbour cells is tested for contact with particle p. In Fig. 1, the level $h = 1$ cells where that search has to be performed (for particle B) are marked in grey.

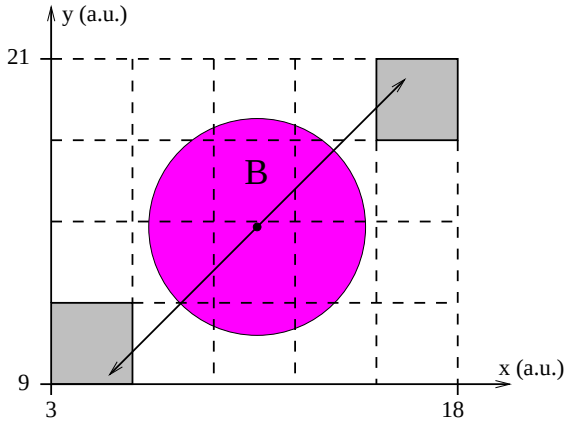

Figure 2: The grey colored cells correspond to cells $\vec{c}^{\,start}$ (left bottom) and $\vec{c}^{\,end}$ (right top) for particle B from the hierarchical grid shown in Fig. (1). The two diagonal vectors with length α, directed to these cells from the center of particle B, are the vectors $\vec{x}_c^{\,-}$ and $\vec{x}_c^{\,+}$ respectively.

To test two particles for contacts, first, the axis-aligned bounding boxes (AABB) of the particles [20] are tested for overlap. Then, for every particle pair which passed this test, the exact geometrical intersection test is applied [2]. Since the overlap test for AABBs is computationally cheaper than for spheres, performing such test first usually increases the performance.

2.3. Summary

The two steps of the algorithm, mapping and contact detection were designed for spherical particles. However, other shapes can also be accommodated using bounding spheres; for an overview of methods to compute a bounding sphere see Ref. [10]. Nevertheless, this can affect the performance when particles are rather elongated.

Parallelization of the algorithm and implementation of periodic boundary conditions are straightforward and can be done in almost the same way as in the Linked-Cell method [21, 22]. Finally, to reduce the memory usage related to storing the cells, we use the hash table approach [8, 10, 12]. This means that the hierarchical grid is not stored explicitly, instead a hash function is used to map only occupied grid cells into a finite 1D hash table.

For bi-disperse particle systems with wide size distributions, for example, $\omega > 10$, the use of a two-level grid can lead to a significant improvement as compared to the Linked-Cell method, as we show below in section 4. Nevertheless, we are interested in finding the optimal grid parameters (L and s_h) for arbitrarily polydisperse particle systems, which will lead to the best performance. In the next section we present a method how to achieve this.

3. Selection of the optimal grid parameters

The algorithm from the last section is applicable to arbitrary systems (inhomogeneous), whereas for the following analysis, we restrict ourselves to almost homogeneous situations. This does not harm/affect the algorithm, only the performance might be sub-optimal. In subsection 3.1 we briefly talk about bi-disperse systems and then focus on polydisperse cases in subsection 3.2.

3.1. Bi-disperse systems

For bi-disperse particle systems the cell sizes of the two-level grid can be easily selected as the two diameters of each particle species. For some situations this may be not as efficient as the use of the single-level Linked-Cell method. In section 4 we show some performance results for bi-disperse size distributions.

3.2. Polydisperse systems

Systems where all the particles' sizes are different we call *polydisperse*. This is the case when a particle sample is drawn from a continuous particle size distribution (PSD), for example using systematic sampling approach [23, 24]. It guarantees a more evenly spread sample, i.e., always includes some of the possibly rare large particles in a sample. For such systems the parameters of the algorithm (the number of levels L and cell sizes s_h) can be chosen in different ways. The performance of the hierarchical grid algorithm then strongly depends on the selected parameters. Consider different numbers of hierarchy levels L. On the one hand, the fewer levels are used, the larger the number of particles per cell. This implies that a larger amount of particles pairs will be found in the contact search, affecting the CPU time dramatically when the system has a large number of small particles. On the other hand, increasing the number of hierarchy levels will decrease the number of particle pairs found in the contact search. This will increase the number of cross-level tests, and hence the number of cells which have to be accessed, negatively affecting the CPU time. To obtain the optimal performance it is important to balance the number of particle pairs found with the number of cells to be checked.

Assume the following hypothesis holds:

Hypothesis 1. *Let m_h be the average number of particles per cell at level h, that is, $m_h = N_h/N_h^c$, where N_h is the number of particles at level h, and N_h^c is the number of cells at this level. Then the optimal distribution of particles by levels satisfies the following condition:*

$$m := m_i = m_j, \ \textit{for all} \ \ i, j \in [1, L]. \tag{6}$$

The detailed discussion of this is in preparation for a future publication.

If the particles are mapped to the levels, such that Eq. (6) is approximated, it can be shown that the CPU time spent for contact detection, T_{CD}, scales as

$$T_{CD} \sim NL(m + K), \tag{7}$$

where K is a constant corresponding to the "overhead" of the algorithm, i.e., the time spent to access cells to be tested, and $m = m(L, \text{PSD})$ is the number of particles per cell. To compute m for a given L and for the particle system at hand, one needs

[2] particles p and q collide only if $\left\|\vec{x}_p - \vec{x}_q\right\| < r_p + r_q$, where $\|\cdot\|$ is Euclidean norm.

to choose cell sizes s_h so that Eq. (6) is approximately satisfied. How to do this is explained in Appendix A. Derivation of Eq. (7) is beyond the scope of this paper. Here, the comparison of the performance results with the prediction of Eq. (7) will be shown.

As obtained from numerical experiments in section 4, the value of K varies in a narrow range $[0.2, \ldots, 0.45]$, depending on the size distribution, and is set to 0.3 with sufficient accuracy.

We propose to use as the optimal number of levels (ONL) the one that minimizes the right hand side of Eq. (7). It must be noted that in general the ONL is relatively small. For example, in the system below with $\omega = 50$ with uniform volume distribution, see Fig. 3(d), ONL = 7.

4. Numerical experiments

The aim of this section is to test the presented algorithm in physically realistic, dilute to dense polydisperse gas- and fluid-like systems. For these we verify experimentally the analytical prediction from Eq. (7). More specific, we use homogeneous and isotropic disordered systems of colliding elastic spherical particles in a cubical box with hard walls. The motion of particles is governed by Newton's second law with a linear elastic contact force during overlap. For simplicity, every particle undergoes only translational motion (without rotation) and gravity is set to zero.

Subsection 4.1 outlines which size distributions are used and why. In subsection 4.2 it is explained how the model systems are prepared. Then, in subsection 4.3, we verify the prediction given by Eq. (7), and also show the performance results for some bi-disperse systems.

4.1. Particle size distributions

The following types of particle size distributions are used: (i) monodisperse, i.e., all sizes are equal; (ii) bi-disperse, i.e., two different sizes, where the volume of all small particles and the volume of all big particles are equal, which is also used in Ref. [25]; (iii) uniform size distribution, i.e., the distribution of radii of the particles is constant; (iv) uniform volume distribution, i.e., the distribution of the volumes of the particles is constant.

We believe that these types of distributions cover the most important cases to check the efficiency of the presented algorithm. Systems with monodisperse size distributions are widely used since kinetic theory predicts their physical behaviour [26, 27, 28], and it is the natural benchmark against which to compare. Bi-disperse size distributions are often used for theoretical models [25, 29, 30], and can often be a good approximation to physically realistic size distributions. Uniform size and uniform volume distributions are selected in order to check the speed-up of the multilevel grid for polydisperse systems with relatively few small particles (uniform size), or rather many small particles (uniform volume). The uniform volume distribution approximates the experimentally obtained size distribution of a concrete mixture [24].

Figure 3 shows the particle systems with different particle size distributions for $N = 125001$ and volume fraction (the ratio between the volume of the particles and the volume of the system) $\nu = 0.62$. Note, that particles of the monodisperse case are ordered (near to walls), since the volume fraction is above 0.55 [28, 31, 32].

4.2. Experimental setup

The systems are prepared in two stages. Starting from a random uniform distribution of points in a cubical box, the radius of the particles grows linearly with time. We use non-overlapping spheres [33], with an Event-Driven code [34], whose growing rate conforms to and conserves a prescribed size distribution. Initial velocities are set randomly in order to keep the system dynamic and random, for details see Ref. [34]. When the target volume fraction is reached, the growth process is stopped.

With the final configuration from the first step, the simulation switches to the relaxation stage with "soft" particles, i.e., particles move according to interparticle forces [1]. The linear elastic normal contact force model is used [35], which leads to a certain contact duration. The integration time step is computed according to the smallest contact duration [35, 36]. At the beginning of this stage the velocities of the particles are scaled in a way that a collision between two of the smallest particles would reach an average maximum overlap of one percent of their radius. We let the simulations run for a few collisions per particle for equilibration before making the measurement of the performance, in order to have contacts (overlaps) between particles in the system.

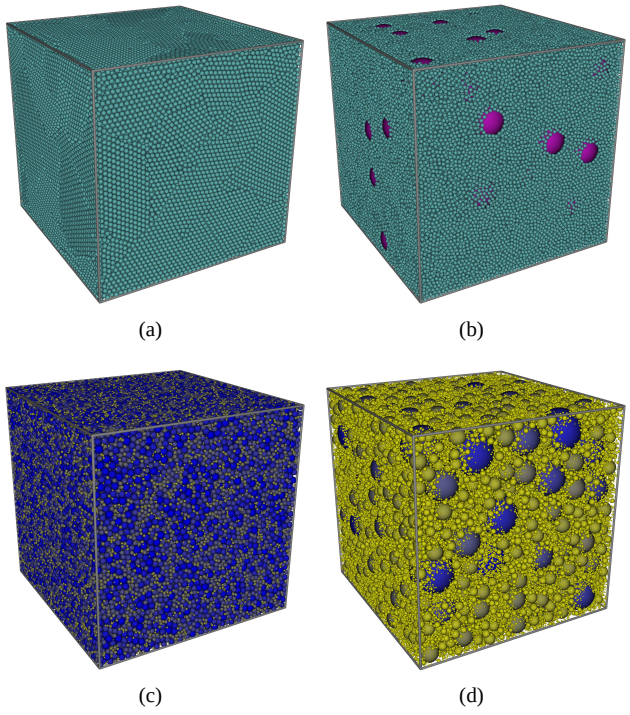

Figure 3: Particle systems with $N = 125001$ and $\nu = 0.62$ with (a) monodisperse size distribution, (b) bi-disperse size distribution, $\omega = 10$, where the volume of all small particles is almost (±2%) equal to the volume of all big particles, (c) uniform size distribution, $\omega = 50$, and (d) uniform volume distribution, $\omega = 50$. Colour is by relative size for the cases (c) and (d).

4.3. Experimental results

To verify the prediction of Eq. (7) we perform two series of experiments, one for varying number of hierarchy levels, and one for different numbers of particles. In the first series, we want to confirm that the multiplier next to N is $L(m + K)$. For this, using a fixed N, we calculate the value of m for each $L \in [1, 50]$ utilizing the method given in Appendix A, and measure the total CPU time of simulations where the hierarchical grid is used with L levels, and the cell sizes s_h are computed in accordance with our hypothesis (1). To present the total CPU time, we use the slowdown factor SF, that is the total CPU time divided by the smallest CPU time for a given system. In Fig. 4 the results of this experiment are shown for systems with uniform size (US) and uniform volume (UV) distributions with $N = 125001$, $\nu = 0.62$ and $\omega = 50$. The analytical prediction (7) is also plotted with $K = 0.3$, scaled in such a way that SF = 1 corresponds to the minimum of the right hand side of Eq. (7). Note that even though the prediction (7) is for CPU time spent only for contact detection, T_{CD}, the total CPU time for fixed N also scales as T_{CD}. This is because the CPU time spent in the force calculation and integration does not depend on the grid parameters used. From the experimental results shown in Fig. 4 it can be seen that: (i) for the system with uniform size distribution the optimal number of levels is $L = 2$, and the speed-up compared to the Linked-Cell method ($L = 1$) is about 30%, so it does not present a major advantage; (ii) in the case of uniform volume distribution the fastest CPU time is achieved using $L = 8$, and the speed-up over the Linked-Cell method ($L = 1$) is of about 220 times. This brings us to the conclusion that for particle systems with relatively few small particles, the use of hierarchical grid algorithm is not essential; however, for the systems with rather many small particles the hierarchical grid algorithm is highly advantageous. Furthermore, it can be seen that the analytical prediction (7) is in very good agreement with the experimental results. We have also performed the same type of experiment for systems with different volume fractions, namely, $\nu = 0.1$ and $\nu = 0.4$, and in all these cases the prediction matches very well with the experimental results.

In the second series of experiments we confirm that the CPU time spent for contact detection scales linearly with the number of particles. Figure 5 shows the total CPU time relative to the monodisperse case, T^{rel}. Polydisperse systems with uniform volume distribution are considered, since it was shown above that for uniform size distributions the use of hierarchical grid algorithm is not essential. The results for bi-disperse systems are also shown, with $L = 2$ and cell sizes equal to the diameters of the each kind of particles. It can be seen that the CPU time scales as $O(N)$, since T_{rel} is approximately constant (±10%) for each type of system, and because the Linked-Cell method for monodisperse particles is $O(N)$. Secondly, the CPU time is close to the monodisperse case ($T^{rel} = 1$), and the largest difference is about 50% in the polydisperse case with $\omega = 10$. Furthermore, the CPU time decreases with increasing ω at constant volume fraction. This is since the mean free path of the system decreases with ω, which determines the number of contact neighbors found, hence the computation time. It must be noted that for large ω we cannot create representative samples with few particles, so we cannot compare all the systems for every N used in the monodisperse case.

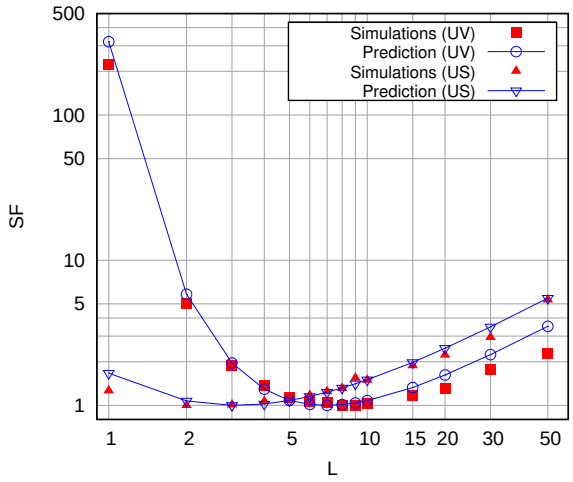

Figure 4: Slowdown factor for different numbers of levels for systems with $N = 125001$, $\nu = 0.62$ and $\omega = 50$ with uniform volume (UV) and uniform size (US) distributions. The prediction of Eq. (7) is used with $K = 0.3$ for plotting solid lines. Note that the data can be obtained only for integer values of L.

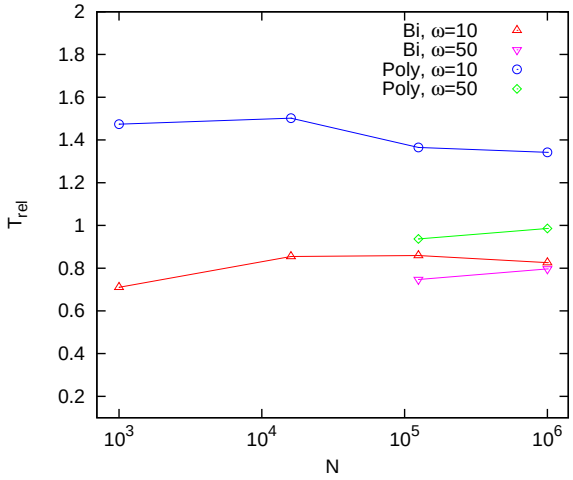

Figure 5: The total CPU time scaled by the total CPU time of a monodisperse system with the same number of particles and volume fraction, simulated for the same number of iterations. We show results for bi-disperse (Bi) and polydisperse uniform volume (Poly) size distributions. Extremely wide size distributions cannot be properly realized for too small N.

5. Summary and Conclusions

A hierarchical grid algorithm for contact detection in systems with arbitrarily polydisperse objects' sizes was developed. DEM simulations were carried out to assess the performance of the algorithm, which contains as adjustable parameters the number of hierarchy levels and the cell sizes at each level. A method to find the optimal parameters for an arbitrary polydisperse size distribution of objects was suggested and confirmed

by numerical experiments. With the optimal parameters our simulations can run orders of magnitude faster than when using the (single level) Linked-Cell method. With this algorithm we are able to simulate objects which have large size ratios with almost the same computational time as in the monodisperse case. With parameters selected using our method the performance of the algorithm scales linearly with the number of particles for any width of the size distribution tested. Our work opens the door to simulate realistic polydisperse systems, making possible a complete new range of simulations.

Acknowledgments

The authors would like to thank S. Gonzalez and A. R. Thornton for helpful discussions. 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 (STW MuST project 10120).

Appendix A. Calculation of $m(L, \mathrm{PSD})$

Assume that particles are sorted by size in increasing order, so that $r_i > r_j$ for $i > j$. We iteratively increase the value of m by $\delta m \ll 1$ starting from zero. Then for every m we distribute particles by levels as following: Starting with the first level, i.e., $h = 0$, and from the smallest particle, i.e., $i = 0$, we allocate the i-th particle at level h. We increase the level index h if for some i the number of particles per cell at current level exceeds m:

$$\frac{N_h(i)}{N_h^c(i)} > m, \tag{A.1}$$

where $N_h(i)$ is the number of particles already allocated at level h (including the particle i) and $N_h^c(i)$ is the number of cells at level h. The number of cells $N_h^c(i)$ is calculated from the size of particle i, e.g., for a 3D cube with size of S, $N_h^c = (0.5S/r_i)^3$. This process is finished when all particles are allocated. Therefore, we obtain the number of levels for a given m. Different m can lead to the same L value. In this case, we select $m(L, \mathrm{PSD})$ with the minimum value, so the last level contains more particles. For numerical experiments in section 4 the use of $\delta m = 10^{-3}$ for the system with $\omega = 50$ with uniform volume distribution and $L = 7$ led to about ±2% variation of the number of particles per cell at different levels.

References

[1] P. A. Cundall, O. D. L. Strack, A discrete numerical model for granular assemblies, Géotechnique 29 (1979) 47–65.

[2] R. A. Gingold, J. J. Monaghan, Smoothed particle hydrodynamics - Theory and application to non-spherical stars, Monthly Notices of the Royal Astronomical Society 181 (1977) 375–389.

[3] J. R. Williams, R. O'Connor, Discrete element simulation and the contact problem, Archives of Computational Methods in Engineering 6 (1999) 279–304.

[4] R. W. Hockney, J. W. Eastwood, Computer Simulation Using Particles, McGraw-Hill, New York, 1981.

[5] M. P. Allen, D. J. Tildesley, Computer Simulations of Liquids, Clarendon Press, Oxford, 1989.

[6] W. Form, N. Ito, G. Kohring, Vectorized and Parallelized Algorithms for Multi-Million Particle MD-Simulation, International Journal of Modern Physics C 4 (1993) 1085–1101.

[7] J. Stadler, R. Mikulla, H. Trebin, IMD: A software package for molecular dynamics studies on parallel computers, International Journal of Modern Physics C 8 (1997) 1131–1140.

[8] T. Iwai, C. Hong, P. Greil, Fast particle pair detection algorithms for particle simulations, International Journal of Modern Physics C 10 (1999) 823–837.

[9] B. Mirtich, Efficient algorithms for two-phase collision detection, in: K. Gupta, A. del Pobil (Eds.), Practical Motion Planning in Robotics: Current Approaches and Future Directions, Wiley, New York, 1998, pp. 203–223.

[10] C. Ericson, Real-Time Collision Detection (The Morgan Kaufmann Series in Interactive 3-D Technology), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[11] H. Mio, A. Shimosaka, Y. Shirakawa, J. Hidaka, Optimum cell condition for contact detection having a large particle size ratio in the discrete element method, Journal of Chemical Engineering of Japan 39 (2006) 409–416.

[12] M. Eitz, G. Lixu, Hierarchical spatial hashing for real-time collision detection, in: IEEE International Conference on Shape Modeling and Applications 2007, Proceedings, pp. 61–70.

[13] K. He, S. Dong, Z. Zhou, Multigrid contact detection method, Physical Review E 75 (2007).

[14] J. F. Peters, R. Kala, R. S. Maier, A hierarchical search algorithm for discrete element method of greatly differing particle sizes, Engineering Computations 26 (2009) 621–634.

[15] A. Munjiza, The Combined Finite-Discrete Element Method, John Wiley & Sons Ltd., 2004.

[16] B. Muth, M.-K. Müller, P. Eberhard, S. Luding, Collision detection and administration for many colliding bodies, in: Proceedings of Discrete Element Methods (DEM) 07, pp. 1–18.

[17] V. Ogarko, S. Luding, Data structures and algorithms for contact detection in numerical simulation of discrete particle systems, in: Proceedings of the 6th World Congress on Particle Technology WCPT6, Nuremberg.

[18] S. Raschdorf, M. Kolonko, A comparison of data structures for the simulation of polydisperse particle packings, International Journal for Numerical Methods in Engineering 85 (2011) 625–639.

[19] T. Ulrich, Loose Octrees, in Mark DeLoura (ed.), Charles River Media, 2000.

[20] M. Moore, J. Wilhelms, Collision detection and response for computer animation, Computer Graphics (SIGGRAPH '88 Proceedings) 22 (1988) 289–298.

[21] D. C. Rapaport, Large-scale molecular dynamics simulation using vector and parallel computers, Computer Physics reports 9 (1988) 1 – 53.

[22] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, Journal of Computational Physics 117 (1995) 1 – 19.

[23] Y. Tillé, Sampling Algorithms, Springer series in statistics, Springer Verlag, New York, NY, 2006.

[24] M. Kolonko, S. Raschdorf, D. Waesch, A hierarchical approach to simulate the packing density of particle mixtures on a computer, Granular Matter 12 (2010) 629–643.

[25] J. M. N. T. Gray, A. R. Thornton, A theory for particle size segregation in shallow granular free-surface flows, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 461 (2005) 1447–1473.

[26] J. P. Hansen, I. R. McDonald, Theory of simple liquids, Academic Press Limited, London, 1986.

[27] S. Miller, S. Luding, Cluster growth in two- and three-dimensional granular gases, Phys. Rev. E 69 (2004) 031305.

[28] S. Luding, Towards dense, realistic granular media in 2D, Nonlinearity 22 (2009) R101–R146.

[29] S. B. Savage, C. K. K. Lun, Particle size segregation in inclined chute flow of dry cohesionless granular solids, Journal of Fluid Mechanics 189 (1988) 311–335.

[30] J. T. Jenkins, D. K. Yoon, Segregation in binary mixtures under gravity, Physical Review Letters 88 (2002) 194301.

[31] A. P. Gast, W. B. Russel, Simple ordering in complex fluids - Colloidal particles suspended in solution provide intriguing models for studying phase transitions, Physics Today 51 (1998) 24–30.

[32] M. D. Rintoul, S. Torquato, Metastability and crystallization in hard-sphere systems, Phys. Rev. Lett. 77 (1996) 4198–4201.
[33] B. D. Lubachevsky, F. H. Stillinger, Geometric properties of random disk packings, J. Stat. Phys. 60 (1990) 561–583.
[34] M. Skoge, A. Donev, F. H. Stillinger, S. Torquato, Packing hyperspheres in high-dimensional euclidean spaces, Phys. Rev. E 74 (2006) 041127.
[35] S. Luding, Cohesive, frictional powders: contact models for tension, Granular Matter 10 (2008) 235–246.
[36] L. E. Silbert, D. Ertaş, G. S. Grest, T. C. Halsey, D. Levine, S. J. Plimpton, Granular flow down an inclined plane: Bagnold scaling and rheology, Physical Review E 64 (2001) 051302.