A. Baietta, F. Maggi
Parallel Packing Code for Propellant Microstructure Analysis
Aerospace Science and Technology, Vol. 46, 2015, p. 484-492
doi:10.1016/j.ast.2015.08.012

Parallel packing code for propellant microstructure analysis

Alessandro Baietta, Filippo Maggi

Aerospace
Science
and
Technology

PII: S1270-9638(15)00256-4
DOI: http://dx.doi.org/10.1016/j.ast.2015.08.012
Reference: AESCTE 3404

To appear in: *Aerospace Science and Technology*

Received date: 15 April 2015
Revised date: 16 July 2015
Accepted date: 31 August 2015

Parallel Packing Code for Propellant Microstructure Analysis

Alessandro Baietta[b,1], Filippo Maggi[b,1,*]

[a] *Politecnico di Milano, 34, via La Masa, 20156 Milan, Italy*

[*] Corresponding author, Tel. +390223998287

Email addresses: baietta.alessandro@gmail.com (Alessandro Baietta), filippo.maggi@polimi.it (Filippo Maggi)

[1] Department of Aerospace Science and Technology, Space Propulsion Laboratory

Parallel Packing Code for Propellant Microstructure Analysis

Alessandro Baietta[b,1], Filippo Maggi[b,1,*]

[b]*Politecnico di Milano, 34, via La Masa, 20156 Milan, Italy*

Abstract

In recent years, packing codes have become a successful alternative to experimental data collection for microstructure investigation of heterogeneous materials. Composite solid rocket propellants are interesting representatives of this category, consisting of a mix of fuel and oxidizer powders embedded in a polymeric binder. Their macroscopic properties are strictly dependent on the peculiar microstructure, which influences mechanical, combustion, as well as physical features. This work addresses algorithm development, validation, and scalability of POLIPack, a parallel packing code based on the Lubachevsky-Stillinger algorithm, developed at the Space Propulsion Laboratory (SPLab) of Politecnico di Milano. The application can reproduce the organization of spheres of any diameter inside a cube with periodic boundary. In addition to the general code description, the paper identifies a collision condition not addressed by the original Lubachevsky's algorithm (here called *back impact*), introduces a novel post-impact handling granting a minimum separation velocity between particles, and presents a parallelization approach based on OpenMP shared memory paradigm. Monomodal and bimodal packs have been compared to experimental data through statistic descriptors and packing maps.

Keywords: packing code, parallelization, propellant, microstructure, OpenMP, Lubachevsky-Stillinger

[*]Corresponding author, Tel. +390223998287

Email addresses: `baietta.alessandro@gmail.com` (Alessandro Baietta), `filippo.maggi@polimi.it` (Filippo Maggi)

[1]Department of Aerospace Science and Technology, Space Propulsion Laboratory

Nomenclature

Acronyms

ANU Australian National University

BDP Beckstead-Derr-Price model

GDF Granular Diffusion Flame

RDF Radial Distribution Function

XCT X-ray Computed Tomography

Greek symbols

Δ Threshold parameter (near-contact number analysis)

ϵ Fraction change parameter

Roman symbols

a Growth rate parameter

D Particle diameter

dr Shell thickness (RDF analysis)

e Restitution coefficient

F Force

f Volumetric packing fraction

$\mathbf{G}$ Relative velocity between centers

m Mass

N Number of particles in the pack

N Total number of particles

$\mathbf{n}$ Unit vector aligned with the line joining two particle centers

n Number of computational steps

$n(r, dr)$ Average number of particles in a shell (RDF analysis)

p Portion of the code that benefits from parallelization (Amdahl's law)

q Number of machines running in parallel (Amdahl's law)

r Radius (RDF analysis)

S Speedup (Amdahl's law)

t Time

V Volume of the pack

$\dot{\mathbf{x}}$ Velocity vector

$\mathbf{x}$ Position vector

Subscripts and superscripts

0 Condition before collision

C Condition at collision

i Ref. to i-th particle

1. Introduction

Composite solid propellants are heterogeneous materials. They are mechanical mixtures of micron-sized powders blended and kept together by a polymer matrix. If a space-averaged analysis is performed on a volume large enough, chemical and physical properties (e.g. composition, density) are uniform. The buildup of properties at the macroscale depends on the microscale and on how the single items arrange each other in the space. Only a close-up observation reveals local nonuniform features introduced by single ingredients. From a propulsion viewpoint, the nominal propellant composition influences the theoretical specific impulse. Nonetheless, global experimental properties such as burning rate, metal agglomeration, or elastic modulus are in part controlled by ingredient morphology. [1].

The heterogeneity of the microstructure was gradually introduced in combustion models, replacing the homogeneous formalism. Initially, one dimensional approaches included a peculiar equation set accounting for the local variability (Summerfield's GDF [2], Hermance [3], Beckstead's BDP [4]). Finally, two- or three-dimensional approaches were developed for laminate or heterogeneous propellant combustion [5–7]. An interesting review on modern methodologies was recently published by Jackson [8]. Multidimensional combustion models for propellants enabled the simulation of micro-to-macroscale combustion physics. End-to-end modeling effort covered different aspects, such as the analysis of the mechanical properties or the propagation of fractures. Also the agglomeration attitude of metalized composite propellants was related to the heterogeneous microstructure. The problem was early addressed by Cohen who developed the pocket concept, which was a geometrically defined region where agglomeration was favored by local conditions. More recently, the concept was generalized to random microstructures via statistical methodologies. [9–12].

Heterogeneous models need reliable information about propellant microstructure. Density and spatial displacement of sphere beads, ellipsoids, powders, or needles, were largely investigated in the past by means of experimental methods with techniques spanning from simple optical analysis to X-ray computed tomography (XCT) [13–19]. Packing codes represent the numerical alternative for the representation of heterogeneous systems. These codes can be implemented using different kinds of algorithms and have the scope to pack items together, into a defined control volume. The shape of the packed objects vary from spheres to arbitrary shapes, depending on the needs [20, 21]. Bandera used a one-by-one placement method, checking for overlaps [22]. Yugong [23] and Maggi [24] applied a collocation strategy based on the minimization of a target function. Webb and Davies used a drop-down technique, with particles randomly released from the top of a virtual container [25]. A time-driven algorithm with inflatable particles and artificial viscosity was applied by Rashkovskii [26]. On the other hand, an event-driven packing algorithm was introduced by Lubachevsky and then used in several codes (e.g. Rocpack [20]). The comparison with experimental tomographic data through spatial statistics for some of the aforementioned techniques have demonstrated that different algorithms do not always generate equivalent microstrucures [27].

This paper describes the features of POLIPack, a parallel implementation of Lubachevsky's algorithm developed at the Space Propulsion Lab (SPLab) of Politecnico di Milano [28–30]. The code, specifically designed for the simulation of propellant microstructure, considers a number of spheres moving in a three-dimensional cubic domain. The items bounce with each other and inflate from a zero-diameter initial condition, till jamming. Periodic boundaries are assumed. The advancement of the algorithm is event-driven and intrinsically concurrent. Tuning, validation, scalability tests, and application to real propellants are presented. Comparison with experimental microstructures was possible thanks to XCT data, kindly provided by Dr. Tomaso Aste from the Australian National University [18, 19].

The paper is organized as follows. Section 2 describes the packing algorithm and the implementation, with emphasis on the collision management. Section 3 reports validation, testing, and sensitivity analyses on growth rate. Section 4 focuses on the parallel implementation of this code and on relevant scalability tests.

2. The packing algorithm

The packing code generates random sphere arrangements in a normalized cubic domain. The side is 2 units long. Opposite faces of the cube are virtually connected, assuring periodic conditions. Despite several strategies have been considered by other authors (solid walls, other domain shapes), the present implementation fulfills the necessity to simulate a bulk of propellant. Otherwise, a solid boundary would have cast its influence on some particle layers from the border.

The particles are represented by spheres which inflate, collide, and move in the domain. The spherical shape grants easier motion handling and collision detection. Despite the use of random shapes would be more adapt to the scope, a sphere can represent adequately also rounded particles with low aspect ratio, typical of most composite propellants. At the beginning of the simulation, each particle is attributed an initial zero-diameter, random position and velocity, and a diameter growth rate proportional to its nominal size. User-defined distributions can be specified arbitrarily. The domain is split in cells for better memory organization and parallelization.

2.1. Event-driven approach

Proper discretization is needed for simulation advancement. The code has to identify and react to events generated by the collision between particles or by interactions with the domain (cell change or border crossing).

In this respect, a time-driven approach, ruled by a time-paced status, can be imprecise and highly expensive from the computational viewpoint [31]. Alternatively, with an event-driven approach the simulation advancement is clocked by a sequence of events. If their identification is efficient and accurate, the corresponding instant of time is derived afterwards.

Identification of collisions or border crossing in the case of particles moving without external forces is simple. The trajectory of the items is a straight line between two collisions. The exact time for the next event is the solution of a linear system (Eq. 1). The reference frame is reported in Fig. 1.

$$m_i \mathbf{x}_i = \mathbf{F}\left(t, \mathbf{x}_i, \dot{\mathbf{x}}_i\right). \quad \mathbf{x}_i = \mathbf{x}_{i0}, \quad \dot{\mathbf{x}}_i = \dot{\mathbf{x}}_{i0}, \quad i = 1, \ldots, N \tag{1}$$

[Figure 1 about here.]

A collision between sphere 1 and 2 happens when the distance of their centers is the sum of their radii. This condition corresponds to the solution of Equation 2, where the collision time t_C is the result.

$$\left\|\mathbf{x}_1\left(t_C\right) - \mathbf{x}_2\left(t_C\right)\right\| = \frac{D_1\left(t_C\right)}{2} + \frac{D_2\left(t_C\right)}{2} \tag{2}$$

In a classical elastic collision between particles of the same diameter, the velocities of the spheres can be divided into components parallel and perpendicular to the line connecting the centers. The transverse components are unchanged by collision, whereas the parallel ones are exchanged. The relationship between velocity across an impact is ruled by Eq. 3 [32].

$$\dot{\mathbf{x}}_1 = \dot{\mathbf{x}}_1^{(0)} - \left(\mathbf{n} \cdot \mathbf{G}^{(0)}\right)(1+e)\,\frac{m_2}{m_1+m_2}\,\mathbf{n} \tag{3a}$$

$$\dot{\mathbf{x}}_2 = \dot{\mathbf{x}}_2^{(0)} + \left(\mathbf{n} \cdot \mathbf{G}^{(0)}\right)(1+e)\,\frac{m_1}{m_1+m_2}\,\mathbf{n} \tag{3b}$$

The computation depends on the pre-impact relative velocity $\mathbf{G}^{(0)} = \dot{\mathbf{x}}_1^{(0)} - \dot{\mathbf{x}}_2^{(0)}$, particle masses m_i, and the restitution coefficient e, which is assumed constant. For $e = 1$, the conservation of kinetic energy through the collision is ensured.

In the case of inflating spheres, the classical theory does not suffice to grant the absence of overlaps. Particle surface features an outward movement with respect to the center, due to the progressive inflation. At collision, additional rebound velocity may be necessary to prevent overlapping. This further contribution depends on diameter growth rate and pre-impact velocity vectors. A simple solution explored by Lubachevsky consisted in adding blindly a constant impulse at every impact but this seemingly naive approach introduces additional rebound velocity even when not needed, thus unnecessarily adding kinetic energy. In this work, the additional rebound velocity matched the one required to ensure the absence of overlaps. The reader should note that even particles with diverging trajectories can collide, if the separating speed is less than the semi-sum of the diameter growth rate. This event may be also a post-collision consequence if the pre-impact relative velocity is low. This specific behavior was not addressed by Lubachevsky and we dubbed it *back impact*. The present code handles this kind of event by forking the execution in two branches, each of them treating the aforementioned cases, and resets rebound velocities to the minimum value ensuring the absence of overlaps.

For more efficient memory use, a queue of N elements is populated by the values of the minimum time lapses to the next event, computed for each particle. A further minimum search identifies the successive event in the whole pack. The code implements a domain decomposition, enabling the collision prediction only among spheres belonging to neighboring cells, otherwise the interaction time is set to infinity. Specific management of particle transfer among the cells or volume border crossing ensures the evolution of the partitioned space.

After an event, the update of the data structure can be synchronous or asynchronous. In the former case, the status of the whole domain is synchronized to the current time once an event is handled. In the latter case, only the parameters of the spheres involved in a collision are updated and a time stamp is associated to the record. This approach is possible because the knowledge of particle location at a known time suffices for the solution of the formulas for linear motion and collision detection. Specific queue data handling is required but consistent saving of computational time is obtained.

2.2. Packing fraction and stopping criteria

The packing fraction f is the part of volume occupied by the spheres. Its experimental value for single-size particle arrangements settle in a limited range. For a random mix of spheres f can span in the range between 0.59 to 0.64, depending on mixing and densification method [17–19]. The reader should remember that the maximum volumetric fraction for a single-size pack is dictated by the close-pack configuration that grants theoretically $f = 0.7405$. This limit cannot be reached just with a random mixing process. Multimodal distributions allow more efficient space utilization since finer particles can arrange within the gaps left by coarse cuts. The packing fraction is correlated to the density of the simulated propellant. In this code each particle belongs to a family (e.g. oxidizer or metal fuel) which is associated to a specific density. It is assumed that the voids between the particles are filled by the binder matrix of known features, unless differently stated.

The Lubachevsky packing algorithm does not come to completion and a stopping criterion is strongly needed. The packing fraction can be used to monitor the progress of the code action. Its value varies run-time, due to sphere inflation, and is dependent also on simulation parameters. For example, higher growth rates reduce the packing capability while slower ones enable higher packing fractions. In this respect, close-pack condition represents an asymptotic state that can be reached only after an infinite simulation time [27]. The simplest possible stopping criterion consists of comparing f against a target packing fraction. This solution is convenient when the reproduction of a known propellant is requested. Alternatively, the definition of a fraction change parameter ϵ, based on the pack history evolution, is a common practice (Eq. 4).

$$\epsilon = \frac{f_{(i+1)\,n} - f_{i\,n}}{f_{(i+1)\,n}} \tag{4}$$

In the formula, i represents the count for ϵ evaluations since the beginning of the run while n is the number of computational steps between two successive checks. Short-period packing fraction oscillations may deceive the stopping decision so in the present code the value of n is assumed ten times the number of particles contained in the pack.

Computation of the next collision time is the code bottleneck [27, 29, 31]. The cell method reduces the number of computations for collision detection, at the expense of managing particle transfers among domain partitions. Sigurgeirsson [31] demonstrated that optimum performance is obtained when cell size and sphere diameter are about the same. Tailoring is easy for a single-size pack. In case of multimodal distributions, domain partitioning is dictated by the biggest particle size. If more items are stored in one cell, loss on performance is observed. The issue becomes more critical when the relative difference between the particle scales increases.

In POLIPack, at least 26 collision predictions must be handled at every iteration, due to the cell splitting of the cubic domain. These computations are independent from each other and are parallelized via shared memory OpenMP paradigm [33]. This programming standard is handled by most of the available commercial or open source compilers and takes advantage of the multi-core architecture available in the latest CPUs. In this kind of problem, a shared memory model simplifies the algorithm because the whole data set is in the scope of each forked process. Drawbacks of this approach consist of the possibility of operation collision on records by competing processes and the limited number of cores in one CPU.

3. Code validation and results

The validation of this code has been accomplished by using a spatial statistical approach based on experimental data available from the literature. Volume-averaged properties (such as the volume fraction) are statistical functions of the first order. They build upon the properties of the microstructure but are not sufficient to identify the correct representation of the matter by a model. Particle-particle reciprocal position can be addressed through descriptors of higher order. In this respect, nearest neighborhood and radial distribution function are also monitored. A detailed comparison has been accomplished against XCT scans of particle packs [2] by Aste and co-authors [18, 19] and on older data obtained via optical packing analysis by Bernard [14], Mason [15] and Gotoh [34].

3.1. Monomodal packs

The statistics of four simulated packs have been compared with two experimental XCT data sets. The effect of particle growth rate and stopping criteria was analyzed. In the packs A-1 and A-4, the final packing fraction was commanded by the fulfillment of the ϵ condition for two different growth rates. Fixed final fraction, matching the property of available experimental data, was imposed in B1 and B2 packs. B1 used a constant growth rate while a Gaussian perturbation of reported standard deviation was added in pack B2. The reader can refer to Table 1 for details. At this stage of the work the size of the particles in the monomodal packs is meaningless since all characterizations are normalized with respect to the final packing diameter.

[Table 1 about here.]

The nearest neighborhood properties are first considered. In a monomodal pack, each sphere cannot be in contact with more than 12 other peers [35]. The average number of contacts is called coordination number (a.k.a. kissing number or contact number). In this type of dynamic simulation, the definition of contacts among spheres can be misleading because only one particle couple is actually touching at every collision detection. Near-contact number is used as an approximation. This is the average number of particles that surround a sphere, which centers are separated by less than $1 + \Delta$ diameters.

[2] XCT data are property of ANU and are used under permission.

Comparison between POLIPack generated packs and experimental data is proposed in fig. 2.

[Table 2 about here.]

[Figure 2 about here.]

A reasonable comparison can be performed only among packs having similar packing fractions since the neighborhood of a particle is more populated for denser packs. Good agreement is observed for POLIPack A-1, A-4 and Gotoh data, which feature a packing fraction between 0.631 and 0.641. If we focus on denser packs, Aste-F and POLIPack B-2 are very similar to each other. For looser packs B-1 and Aste-D it appears that the immediate neighborhood of the experimental particle assembly is denser with respect to the simulated one, while the opposite trend is observed at a longer range. In this respect, the nature of the packing algorithm and the problem of contact definition may be emphasized for loose packs. It is worth noting that experimental data often do not agree with each other. For example, Bernal's data seem to overestimate the values assessed by Gotoh and Aste while Mason's data are more representative but the information about packing fraction is missing.

The Radial Distribution Function (RDF) is a statistical descriptor of the second order. It characterizes the reciprocal position of spheres, identifying repetitive structures in assemblies, or clusters in multimodal blends [12]. It does not supply a directional information.

$$g(r, dr) = \frac{V}{N} \frac{n(r, dr)}{4\pi r^2} \tag{5}$$

The function (Eq. 5) compares the average particle intensity of the pack N/V with the same information computed at a given distance r from each particle and globally averaged. The parameter dr is the thickness of the shell used for the construction of function $n(r, dr)$. RDFs of available data are contrasted in fig. 4, normalizing the horizontal axis with respect to the particle diameter. The comparison reported in fig. 4(a) shows a good agreement between experimental data (Aste F) and simulation A-4 from POLIpack, both featuring a packing fraction of 0.64. Some peculiar structures can be identified in the particle arrangement. The peak at the distance of 2 diameters represents the presence of three aligned spheres; another peak at $\sqrt{3}$ is generated by three spheres in contact, whose centers form an equilateral triangle. Neither experimental nor simulated data display a tetrahedral configuration, whose fingerprint is a peak at $\frac{2}{3}\sqrt{6}$. Some possible configurations are illustrated in fig. 3. The RDF of POLIPack A-4 is characterized by higher peaks with respect to Aste F, probably due to the uncertainty of sphere center identification from the tomographic experiment, which is not known. In this respect, POLIpack B-2 introduced some level of uncertainty by sampling the diameter growth rates from a Gaussian distribution. More similar results with smoothed peaks are obtained (fig. 4(b)).

[Figure 3 about here.]

[Figure 4 about here.]

Also for packs Aste-D, POLIPack-B1, and POLIPack-A4 the RDF analysis shows good agreement even though peaks from experimental data are not as evident as in simulated packs. In these cases, particles are looser with respect to the previous examples: Rattlers (particles with some room of movement) may be present and their location can generate peak smoothing (fig. 4(c) and fig. 4(d)).

3.2. Growth rate and rebound velocity

Previous works have demonstrated that the result of Lubachevsky's algorithm is influenced by the choice of the particle growth rate. For example, the final packing fraction in Rocpack code could be tuned by the choice of this parameter [20]. This feature is positive when the target packing fraction is known because it enables a wider simulation range. On the other side, the choice of the proper growth rate for an unknown pack might be problematic and can lead to result variability. In the present work we have discussed about the treatment of after-collision velocity. With respect to the original Lubachevsky's algorithm, it is possible to limit the growth of the kinetic energy by limiting the increment of rebound velocity. This feature was tested for POLIPack, using an assembly of 3000 single-size spheres. The diameter growth rate a was varied from 0.01 to 100 on a logarithmic increment. The stopping criterion was $\epsilon = 10^{-7}$, across a number of iterations, arbitrarily set to 30000. With respect to the standard code (version A), three different variants of minimum separation velocity between sphere centers after a collision ($\dot{\mathbf{x}}_{min}$) and restitution coefficient (e) have been tested, as reported in tab. 3. The results are presented in fig. 5, along with the outcomes of Rocpack reference code. Each point in the plot is the average of three runs with different initializations.

[Table 3 about here.]

[Figure 5 about here.]

The resulting packing fractions are well centered in the range obtained experimentally by McGeary [17]. Specifically, Code B tends to $f = 0.625$, which was found for an experimental assembly of steel spheres. The sensitivity against the growth rate a is reduced with respect to the reference code and plays a significant role only for higher packing fractions. The variation of the post-collision strategy operates a generic translation of the whole set of results. Code C and D do not demonstrate significant differences with each other and set a packing fraction of about 0.61, in the testing range and for the assumed stopping criterion. Code version A is capable of generating the tightest packs for different conditions while intermediate results are obtained from version B of the algorithm. The difference between code A and B demonstrates the influence of post-collision velocity setup. The higher is $\dot{\mathbf{x}}_{min}$, the higher is the final packing fraction. The influence vanishes for lower growth rates. The same trend is observed for the increment of the restitution coefficient e, between code C and B. No influence is visible, below the threshold of $e = 0.5$. Resulting microstructures were verified to be compliant with the experiments.

3.3. Binary packs

The packing of a powder blend depends on the sphere distribution and on the method to generate the particle assembly ([36–38]). From the experimental standpoint, McGeary reported interesting results and processing details for binary packings, documenting data for different coarse-to-fine diameter and quantity ratios [17]. Densification was achieved mainly by shaking and was stopped following a minimum volume criterion. From a numerical viewpoint, the stopping criterion of the Lubachevsky's algorithm is still an open issue, if the final packing fraction is not known. The progression of particle inflation is monitored through the parameter ϵ but its arbitrary choice can cause data fluctuation. Figure 6 reports the case of a bimodal pack containing 80% of coarse and 20% of fine cuts, having diameter ratio 6.5 : 1. The volumetric packing fraction increases beyond the experimental value while ϵ tends to a quasi-asymptotic trend. The dashed line represents the experimental value for the specific condition and is matched for $\epsilon \approx 5 \times 10^{-4}$.

[Figure 6 about here.]

Two series of packs made by 20000 spheres were generated varying the coarse-to-fine volume fraction and keeping constant the coarse-to-fine diameter ratio. Growth rate of 0.2 and $\epsilon = 5 \times 10^{-4}$ were chosen. Figures 7(a) and 7(b) report POLIPack results, McGeary's experimental data, and Rocpack curves (reference code). It is worth remembering that growth rates may not be directly comparable between the two codes. Also the plots report two lines representing the ideal packing limits, given by the superimposition of two monomodal random packs.

[Figure 7 about here.]

The selected stopping criterion works quite well for the pack with diameter ratio 6.5:1. Both POLIPack and the reference code underestimate the maximum packing fraction of about 2%, suggesting that it is not a code-related issue. Experimental and numerical data agree in the rest of the map. The modeled packs with 3.4:1 diameter ratio do not show the same agreement with experimental data. Both POLIPack and Rocpack with a = 0.2 overestimate McGeary's maps for a fine fraction lower than 40%. Above this limit, full agreement is obtained. This behavior is not yet understood and should be matter of further investigation. The change of prediction capability after growth rate variation in Rocpack suggests that collision handling can play a significant role but introduces the difficulty in the selection of the correct value for a. For POLIPack, a preference is given to the identification of a proper stopping criterion ϵ, rather than a variation in growth rate, since the proposed implementation features a reduced sensitivity of the code on a. In any case, the reliable representation of a wide family of packs seems to be possible if the final packing fraction is already known. Rather, the preliminary choice of an ϵ does not ensure that the code stops at the desired fraction for a wide set of fine-to-coarse particle combinations, as it was observed in Figures 7(a) and 7(b). This fact opens to the identification of a different converging criterion which does not ground only on the evolution of the packing fraction.

4. Scalability test

An important innovation introduced by the present code consists in the approach to parallelization of the Lubachevsky's algorithm. The OpenMP shared memory paradigm simplifies code structure and eliminates inter-process data transfer because variable space can be shared among the running instances. As drawbacks, the number of parallel processes is upper limited by the independent threads available for a machine and by the total amount of memory. The latter problem is not an issue since million-sized packs can be stored in some tens of megabytes. The former problem is more stringent and is correlated to the number of cores of the processor(s) installed in the same mainboard. Some technologies (such as Intel Hyper-Threading®) may enable running speedup even with multiple independent threads per each core but the scalability is not full. Reported tests have been performed on a dual core Hyper-Threading enabled Intel Core i5-480M machine with 4 GB DDR3 memory and Windows 7 operating system. With this specific configuration, significant results can be obtained for runs using up to 4 processes.

The parallelization acts on the domain decomposition, mentioned at the beginning of Section 2. Collision detection between the content of a cell and of its surrounding peers can be constructed as an independent process and forked into parallel runs. The code can control the number of cells generated in the domain and, indirectly, the number of particles per cell by changing the total particle number. For the following code runs, the domain was split into 64 cells. Scalability tests were performed for two and four threads, varying the number of particles, and reporting the ratio between serial and parallel wall-clock time needed to reach a given number of collision predictions (Figure 8). In case of monomodal packs, benefits from parallelization are evident for more than 500 particles per cell. Scalability works quite well: two processes almost half the execution time while reduction of 3.5 times is observed for four threads (using Hyper-threading). This latter speedup may not be due to parallelization reasons only, considering literature data on the subject [39]. Memory and caching improvements may have occurred but it is hard to say.

[Figure 8 about here.]

[Figure 9 about here.]

Parallelization is not always convenient. In fig. 9 a magnification of the zone between 10 and 100 spheres per cell is presented. The single-threaded version results to be faster when there are less then about 40 particles in a cell. In this case OpenMP overhead, needed to fork the process, is more time consuming than the code executed in each fork, killing performance gains. From an ideal viewpoint, the speedup should be 2 or 4 for two-thread and four-thread runs, respectively. Amdahl about fifty years ago defined a law for the computation of the speedup capability for a code running in parallel on q equivalent machines. Assuming that only the portion p of the code could benefit from the parallelization, he noted that the speedup s was governed by Eq. 6 [40].

$$s = \frac{1}{(1-p) + \frac{p}{q}} \tag{6}$$

The complexity of current multicore machines makes more difficult the definition of a rigorous scalability law since the architecture of the central processing unit should be contemplated in the analysis [41]. Moreover, it is hard to identify the portion of the code subjected to effective parallelization for a complex C++ code, unless a case-by-case profiling is performed. For the purpose of this research, an empirical evaluation of parallelization efficacy with respect to the ideal case was obtained by deriving from the Amdahl's equation the coefficient p (Eq. 7).

$$p = \frac{\frac{1}{s} - 1}{\frac{1}{q} - 1} \tag{7}$$

The parameter ranges between 0 and 1, tending to unity only if the code can fully exploit the parallelization. In this latter case, for $p = 1$ the ideal speedup coincides with the number of parallel branches ($s = q$) while $p < 1$ highlight the presence of overheads. Figure 10 reports the performance of the parallel algorithm for different computational cases. The trend shows that the higher is the number of particles in each computational cell, the better is the efficacy of the parallelization. Results show an asymptotic behavior, overcoming $p = 0.9$ for cell content above 1000 particles. It is interesting to note that the efficacy of a two-thread process is lower than a four-thread case. The difference tends to disappear for increased number of particles and may be caused by the peculiar architecture of the machine used for the simulation. The plot does not report data for a number of particles lower than 50 since $p < 0$ is obtained, meaning that parallelization is not convenient.

[Figure 10 about here.]

5. Conclusions

The implementation of POLIPack, a packing code based on the Lubachevsky-Stillinger algorithm was presented in this paper, describing governing framework, implementation, parallelization, and validation tests. The application demonstrated the capability to generate three-dimensional assemblies of polydisperse spheres in a periodic domain, having comparable properties with respect to experimental data. In addition to the standard Lubachevski algorithm, this code included the handling for a new collision, not considered by the original methodology, and derived a new post-collision handling concept, based on a criterion of minimum separation velocity between particles. Parametric tests have demonstrated that the present code features lower sensitivity on the particle growth rate with respect to other comparable implementations. The code was parallelized using a shared memory paradigm which ensured a simpler code and demonstrated good scalability properties on architecture typically used for laptop and desktop computers. A parametric analysis on packing fraction data and comparison with experiments suggested that a common methodology for the identification of a stopping criterion is still lacking and, up to now, proper tuning is required on controlling parameters, namely, particle growth rate and packing fraction variation. In this respect, packing fraction evolution, growth rate of kinetic energy, or even progress of microstructure properties should be mapped in monomodal and multimodal packing procedures. The current implementation of the code features limited flexibility on domain type and particle shape. Whereas the extension to other domains (cuboids or cylinders) or addition of external forces (e.g. virtual shaking or gravity) result in a quite straightforward implementation within the bounds of the current framework, the collision strategy is not suitable for non-spherical particles and must be reconsidered.

6. Acknowledgements

The authors acknowledge Dr. Tomaso Aste from Australian National University, who supplied the experimental data for validation. The reference code Rocpack is property of the University of Illinois at Urbana Champaign and was developed at the Center for Simulation of Advanced Rockets (CSAR).

[1] N. Kubota. *Propellants and Explosives: Thermochemical Aspects of Combustion*. 2nd edition, Wiley-VCH, Weinheim, Germany, 2006.

[2] J. A. Steinz, P. L. Stang, and M. Summerfield. The burning mechanism of ammonium perchlorate-based composite solid propellants. *Aerospace and Mechanical Sciences Report No. 830*, Princeton University, Princeton, NJ, USA, DTIC Ref. AD688944, 1969.

[3] C. E. Hermance. A model of composite propellant combustion including surface heterogeneity and heat generation. *AIAA J*, 4(9):1629–1637, 1966. doi:10.2514/3.55284

[4] M. W. Beckstead, R. L. Derr, and C. F. Price. A model of composite solid-propellant combustion based on multiple flames. *AIAA J*, 8(12):2200–2207, 1970. doi:10.2514/3.6087

[5] T. L. Jackson and J. Buckmaster. Heterogeneous propellant combustion. *AIAA J*, 40(6):1122–1130, 2002. doi:10.2514/2.1761

[6] F. Miccio, and . Numerical modeling of composite propellant combustion. In *Proc Combust Inst*, 27(2):2387–2395, 1998. doi:10.1016/S0082-0784(98)80090-8

[7] P. Genevieve and M. Q. Brewster. Modeling combustion in laminate propellants with fine ammonium perchlorate. AIAA Paper, No. 2002-0781, 2002. doi:10.2514/6.2002-781

[8] T. L. Jackson. Modeling of heterogeneous propellant combustion: A survey. *AIAA J*, 50(5):993–1006, 2012. doi:10.2514/1.J051585

[9] N. S. Cohen. A pocket model for aluminum agglomeration in composite propellants. *AIAA J*, 21(5):720–725, 1983. doi:10.2514/6.1981-1585

[10] S. A. Rashkovskii. Role of the structure of heterogeneous condensed mixtures in the formation of agglomerates. *Combust Explos Shock Waves*, 38(4):435–445, 2002. doi:10.1023/A:1016211215981

[11] S. Gallier. A stochastic pocket model for aluminum agglomeration in solid propellants. *Propell Explos Pyrot*, 34(2):97–105, 2009. doi:10.1002/prep.200700260

[12] F. Maggi, L. T. De Luca, A. Bandera. Pocket model for aluminum agglomeration based on propellant microstructure. *AIAA J*, in press, 2015. doi:10.2514/1.J053992

[13] J. D. Bernal, I. A. Cherry, J. L. Finney, and K. R. Knight. An optical machine for measuring sphere coordinates in random packings. *J Phys E*, 3(5):388–390, 1970. doi:10.1088/0022-3735/3/5/312

[14] J. D. Bernal and J. Mason. Packing of spheres: Co-ordination of randomly packed spheres. *Nature*, 188(4754):910–911, 1960. doi:10.1038/188910a0

[15] G. Mason. Radial distribution functions from small packings of spheres. *Nature*, 217(5130):733–735, 1968. doi:10.1038/217733a0

[16] G. Mason and W. Clark. Fine structure in the radial distribution function from a random packing of spheres. *Nature*, 211(5052):957, 1966. doi:10.1038/211957a0

[17] R. K. McGeary. Mechanical packing of spherical particles. *J Am Ceram Soc*, 44(10):513–522, 1961. doi:10.1111/j.1151-2916.1961.tb13716.x

[18] T. Aste, M. Saadatfar, and T.J. Senden. Geometrical structure of disordered sphere packings. *Phis Rev E*, 71(6), 2005. doi:10.1103/PhysRevE.71.061302

[19] T. Aste, T. J. Senden, M. Saadatfar, and A. Sakellariou. Investigating the geometrical structure of disordered sphere packings. *Physica A*, 339(1–2):16–23, 2004. doi:10.1016/j.physa.2004.03.034

[20] G. M. Knott, T. L. Jackson, and J. Buckmaster. Random packing of heterogeneous propellants. *AIAA J*, 39(4):678–686, 2001. doi:10.2514/2.1361

[21] D. S. Stafford and T. L. Jackson. Using level sets for creating virtual random packs of non-spherical convex shapes. *J Comput Phys*, 229(9):3295–3315, 2010. doi:10.1016/j.jcp.2010.01.003

[22] A. Bandera. *Combustion of metalized solid rocket propellants and motor performance*. PhD thesis, Politecnico di Milano, Milan, Italy, 2009.

[23] Y. Wu, Z. Fan, and Y. Lu. Bulk and interior packing densities of random close packing of hard spheres. *J Mater Sci*, 38(9):2019–2025, 2003. doi:10.1023/A:1023597707363

[24] F. Maggi. *Heterogeneity effects on composite solid rocket propellants*. PhD thesis, Politecnico di Milano, Milan, Italy, 2006.

[25] M. D. Webb and I. L. Davis. Random particle packing with large particle size variations using reduced-dimension algorithms. *Powder Technol*, 167(1):10–19, 2006. doi:10.1016/j.powtec.2006.06.003

[26] S. A. Rashkovskii. Structure of heterogeneous condensed mixtures. *Combust Explos Shock Waves*, 35(5):523–531. 1999, doi:10.1007/BF02674497

[27] F. Maggi, S. Stafford, T. L. Jackson, and J. Buckmaster. Nature of packs used in propellant modeling. *Phys Rev E*, 77(4), 2008. doi:10.1103/PhysRevE.77.046107

[28] B. D. Lubachevsky. How to simulate billiards and similar systems. *J Comput Phys*, 94(2):255–283, 1991. doi:10.1016/0021-9991(91)90222-7

[29] B. D. Lubachevsky and F. H. Stillinger. Geometric properties of random disk packings. *J Stat Phys*, 60(5–6):561–583, 1990. doi:10.1007/BF01025983

[30] A. Baietta. *Development of random packing code for solid propellant microstructure simulation*. M.Sc. thesis, Politecnico di Milano, Milan, Italy, 2012.

[31] H. Sigurgeirsson, A. Stuart, and W. Wan. Algorithms for particle-field simulations with collisions. *J Comput Phys*, 172(2):766–807, 2001. doi:10.1006/jcph.2001.6858

[32] C. T. Crowe, T. Yutaka, and M. Sommerfeld. *Multiphase Flows with Droplets and Particles*. CRC Press, 1st edition, Boca Raton, FL, USA, 1998.

[33] Anon., *OpenMP Application Program Interface*, version 2.5 edition, 2005.

[34] K. Gotoh and J. L. Finney. Statistical geometrical approach to random packing density of equal spheres. *Nature*, 252(5480):202–205, 1974. doi:10.1038/252202a0

[35] J. Leech. The problem of the thirteen spheres. *Math Gaz*, 40(331):22–23, 1956. doi:10.2307/3610264

[36] A. Rosato, K. J. Strandburg, F. Prinz, and R. H. Swendsen. Why the Brazil nuts are on top: size segregation of particular matter by shaking. *Phys Rev Lett*, 58(10):1038–1040, 1987. doi:10.1103/PhysRevLett.58.1038

[37] A. P. J. Breu, H.-M. Ensner, C. A. Kruelle, and I. Rehberg. Reversing the Brazil–nut effect: competition between percolation and condensation. *Phys Rev Lett*, 90(1):014302/1–014302/3, 2003. doi:10.1103/PhysRevLett.90.014302

[38] T. Aste and D. L. Weaire. *The Pursuit of Perfect Packing*. CRC Press, 2nd edition, Boca Raton, FL, USA, 2008.

[39] W. Magro, P. Petersen, and S. Shah. Hyper-threading technology: Impact on compute-intensive workloads. *Intel Tech J*, 6(1):1–9, 2002.

[40] G. M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. *AFIPS Conf P* , 30:483–485, 1967.

[41] M. D. Hill, M. R. Marty. Amdahl's law in the multicore era. *Computer* , 2008(7):33–38, 2008.

List of Figures

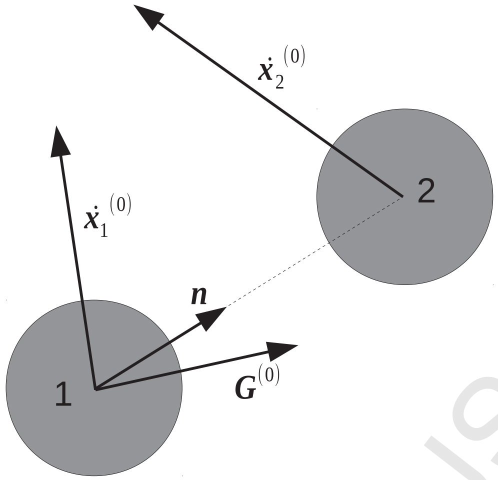


Figure 1: Reference system for particle collision

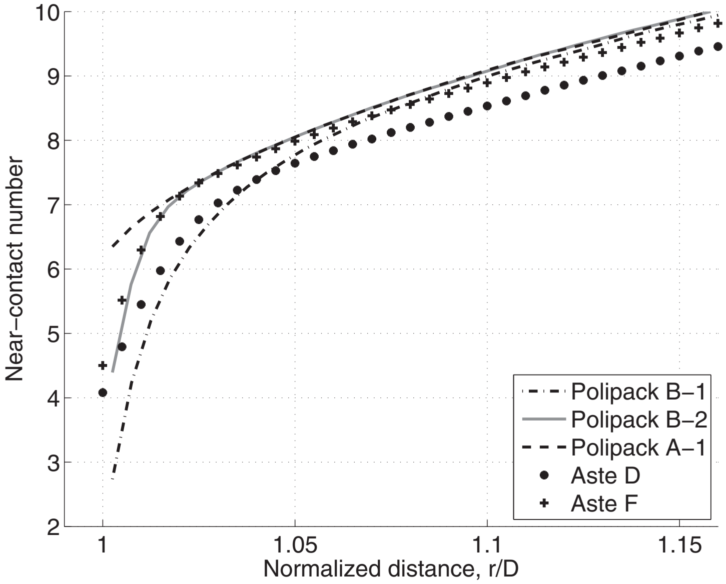


Figure 2: Comparison of the average near contact number as function of normalized distance.

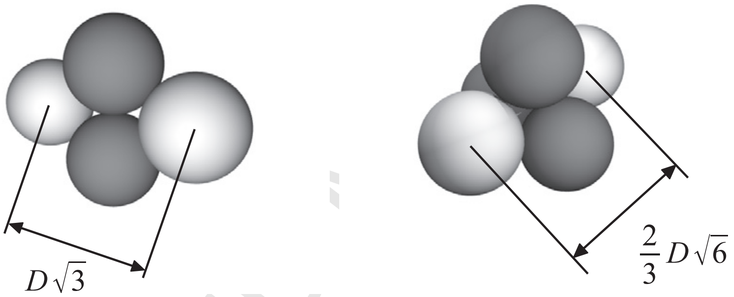
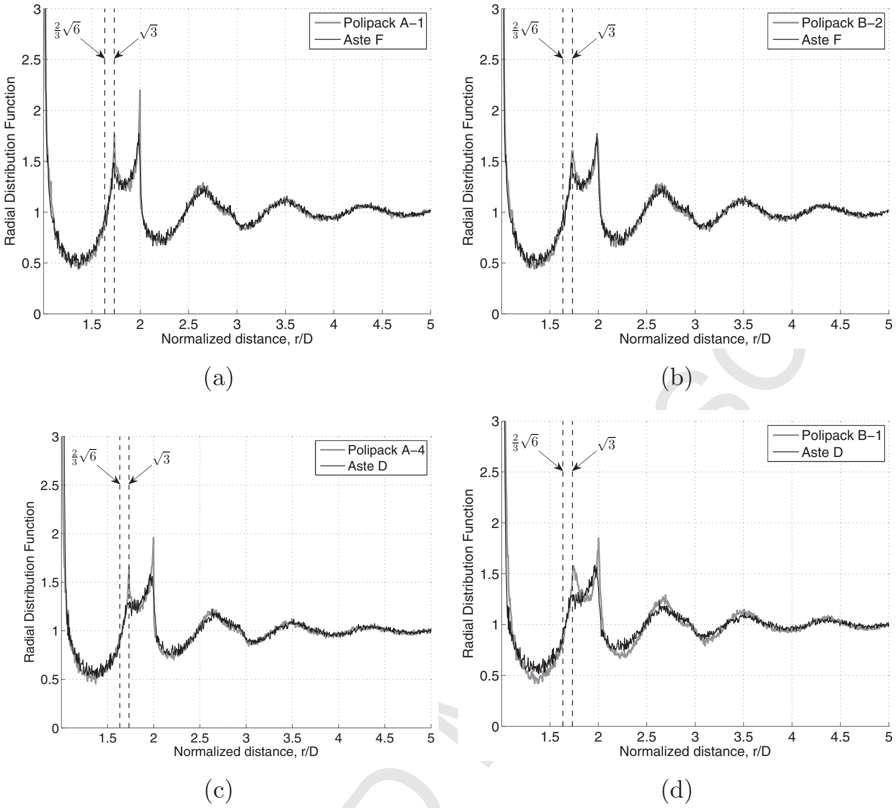


Figure 3: Possible sphere arrangements and the resulting distances. On the left, four spheres lie on a plane, the centers form an equilateral triangle; on the right side, the darker spheres form a regular tetrahedron with each of the lighter ones.



(a)

(b)

(c)

(d)

Figure 4: Radial Distribution Function; comparison between experimental data from Aste [18, 19] and simulated packs (see Table 1 for further details).

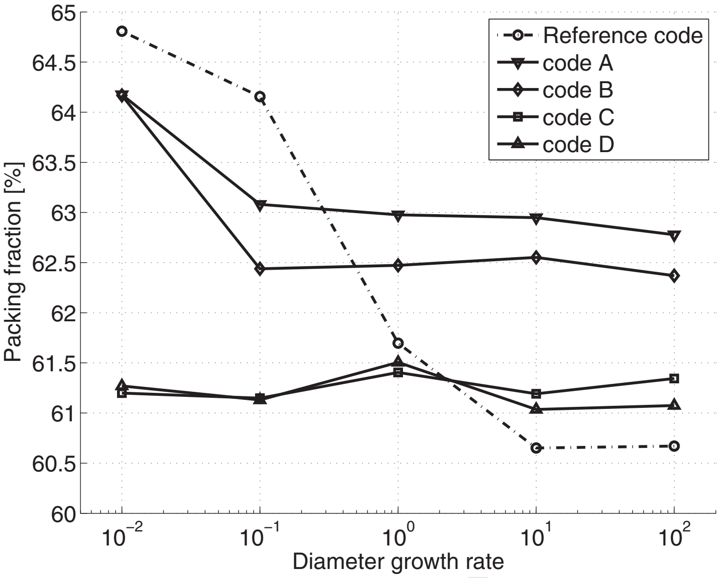


Figure 5: Dependence of packing fraction on diameter growth rate in single-size packs. Code A represents the baseline. Reference code is “Rocpack”

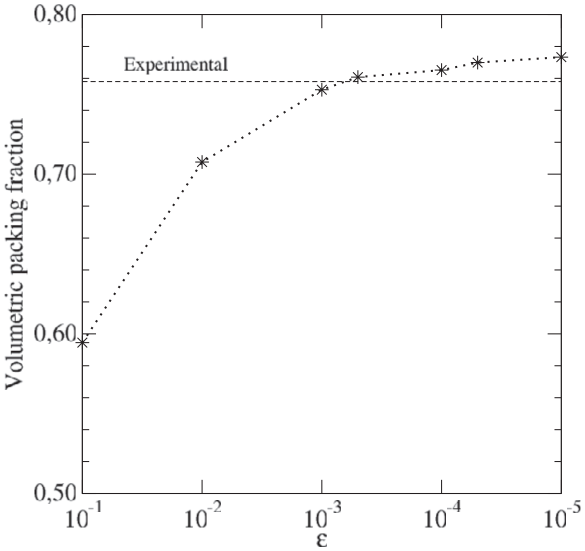
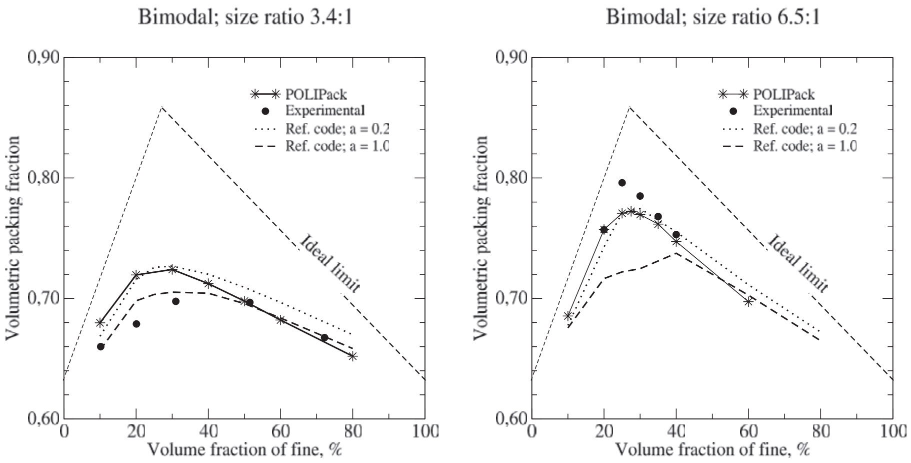


Figure 6: Evolution of volumetric packing fraction versus ϵ parameter for a bimodal packing having a coarse-to-fine ratio of 80/20 and a diameter ratio of 6.5:1. Direction of the horizontal axis is inverted. Packing progression runs from left to right.



(a) Diameter ratio: 3.4:1

(b) Diameter ratio: 6.5:1

Figure 7: Bimodal packing maps. Data from experiments and Rocpack reference code are included [17, 27].

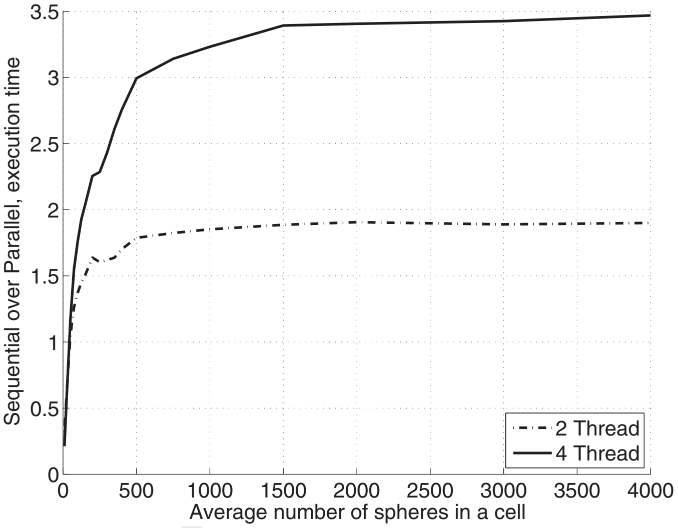


Figure 8: Scalability tests of POLIPack. Execution time of the parallel version is expressed as fraction of the sequential one.

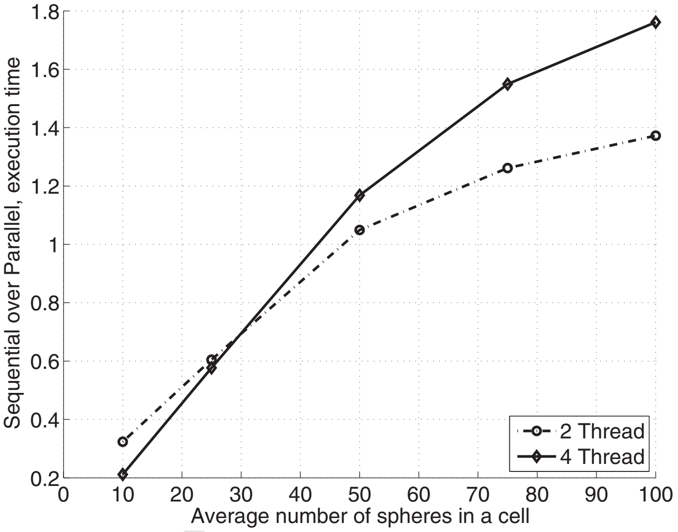


Figure 9: Same as fig. 8; transition zone magnified. For more than 30 spheres per cell the OpenMP version of the program results to be faster than the sequential one.

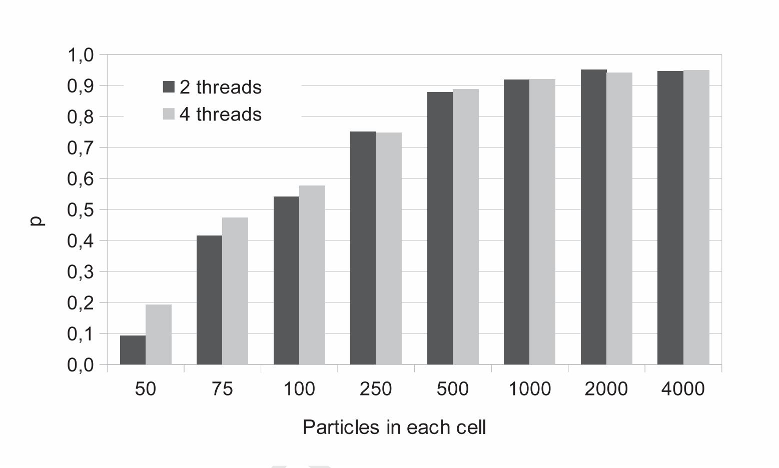


Figure 10: Performance of parallelization for different computational cell content

List of Tables

Label	a	f	Source	Note
POLIpack A-1	0.01	0.641	Simulated	Stop $\epsilon < 10^{-7}$
POLIpack A-4	10	0.631	Simulated	Stop $\epsilon < 10^{-7}$
POLIpack B-1	0.01	0.626	Simulated	Stop $f <= 0.626$
POLIpack B-2	0.01 ± 10^{-4}	0.640	Simulated	Stop $f <= 0.640$

Table 1: Packs used for the statistical validation of the microstructure

Source	f	N	Near-contact number $1.02\,D$	 $1.05\,D$	 1.10
Bernal-LRP	0.60	420	N.A.	7.1	N.A.
Bernal-DRP	0.62	476	N.A.	8.5	N.A.
Mason	N.A.	536	6.8	8.0	8.9
Gotoh	0.636	7934	7.05	8.0	9.0
POLIpack A-1	0.641	10000	7.17	8.05	9.09
POLIpack A-4	0.631	10000	7.07	7.86	8.79
POLIpack B-1	0.626	10000	6.11	7.78	9.00
POLIpack B-2	0.640	10000	7.13	8.05	9.08
Aste D	0.626	35000	6.43	7.65	8.53
Aste F	0.640	35000	7.13	7.99	8.89

Table 2: Near-contact number; comparison between POLIPack results (using $e = 1$) and experimental data [14, 15, 18, 19, 34]. Values for distances of $1.02\,D$, $1.05\,D$ and $1.10\,D$ are reported.

Code	$\dot{\mathbf{x}}_{min}$	e
A (baseline)	$2.00 \cdot (a_0)$	1.0
B	$1.02 \cdot (a_0)$	1.0
C	$1.02 \cdot (a_0)$	0.5
D	$1.02 \cdot (a_0)$	0.0

Table 3: Different combination of $\dot{\mathbf{x}}_{min}$ and e used in the collision function.