Computer Physics Communications 196 (2015) 247–254

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

A geometric approach to identify cavities in particle systems

Evangelos Voyiatzis *, Michael C. Böhm, Florian Müller-Plathe

Eduard-Zintl-Institut für Anorganische und Physikalische Chemie and Center of Smart Interfaces, Technische Universität Darmstadt,
Alarich-Weiss-Strasse 4, 64287 Darmstadt, Germany

ARTICLE INFO

Article history:
Received 20 January 2015
Received in revised form
15 June 2015
Accepted 17 June 2015
Available online 9 July 2015

Keywords:
82.20. Wt
82.35. Lr
83.10. Rs
Accessible volume
Cavity size
Cavity shape
Python
Delaunay triangulation

ABSTRACT

The implementation of a geometric algorithm to identify cavities in particle systems in an open-source python program is presented. The algorithm makes use of the Delaunay space tessellation. The present python software is based on platform-independent tools, leading to a portable program. Its successful execution provides information concerning the accessible volume fraction of the system, the size and shape of the cavities and the group of atoms forming each of them. The program can be easily incorporated into the LAMMPS software. An advantage of the present algorithm is that no *a priori* assumption on the cavity shape has to be made. As an example, the cavity size and shape distributions in a polyethylene melt system are presented for three spherical probe particles. This paper serves also as an introductory manual to the script. It summarizes the algorithm, its implementation, the required user-defined parameters as well as the format of the input and output files. Additionally, we demonstrate possible applications of our approach and compare its capability with the ones of well documented cavity size estimators.

1. Introduction

The accessible volume fraction and cavity size distribution in soft matter systems determine many of their static and dynamic properties [1–8]. Among others, the size and shape of cavities in polymers have a strong impact on the permeability and solubility of small molecules [1,2]. Attempts to describe the liquid–glass transition in amorphous materials by modifications of the free volume have been made [3]. One of the dominant mechanisms leading to failure in amorphous polymers under deformation is crazing [4–6], i.e. the formation of large cavities which greatly reduce the cohesive forces in these materials. It belongs to the common knowledge that the diffusion of small molecules in polymer matrices can be described by a series of hopping processes between neighboring cavities [7,8].

Several algorithms to identify cavities in molecular simulations have been described in the literature [7,9–13]. They can be divided into two main categories. The first one makes use of geometric algorithms to describe the molecular system in terms of an equivalent hard-sphere one and neglects all energetic interactions among the particles. Then, the analysis relies either on a Delaunay space tessellation or a space discretization using for instance cubic grids [14,15]. The second category employs the energetic interactions between a probe particle and the atoms of the system to detect the cavities [16,17]. The majority of these techniques assumes a spherical cavity shape [10]. An advantage of the first set of methods is that no assumption on the shape of the cavities is made.

In this paper, we reconsider a geometric algorithm to determine the cavity size distribution and the accessible volume fraction in particle systems. The proposed cavity identification procedure employs spherical probes. Non-spherical cavity shapes are not determined directly, i.e. by employing non-spherical probes, but indirectly by using spherical probe particles and calculating several cavity shape parameters, such as the asphericity and acylindricity elements. Its implementation into a portable open-source python script which can be integrated into LAMMPS [18] is presented. The source code is distributed at the Pizza.py toolkit website (http://pizza.sandia.gov/). A direct adaptation of the cavity identification code to other Molecular Dynamics or Monte Carlo software programs is feasible. It requires only a simple modification of the parsing module of the input files. An alternative approach would be the employment of utilities, such as the built-in interfaces of LAMMPS or the use of Topotools [19], to convert between different file formats. The influence of the size of the probe particle on the estimated properties is discussed for an unentangled monodisperse polyethylene melt.

* Corresponding author. Tel.: +49 615116 6537; fax: +49 615116 6526.
E-mail address: e.voyiatzis@theo.chemie.tu-darmstadt.de (E. Voyiatzis).

http://dx.doi.org/10.1016/j.cpc.2015.06.011
0010-4655/© 2015 Elsevier B.V. All rights reserved.

Table 1
List of all user-defined parameters in the python cavity identification application.

Parameter	Description
NumConfs	Integer variable defining how many configurations from the configuration file will be processed, starting from the first
maxMC	Integer variable determining the number of insertions in the Monte Carlo accessible volume calculation for each Delaunay tetrahedron
ClusteringRadius	Real variable specifying the probe radius
UnoccupiedVolumeInfo	Integer variable controlling whether the unoccupied volume of each tetrahedron will be stored in an external file (value of 0) or not (any other value)
AtomsPerCavityInfo	Integer variable controlling whether the volume and the atoms forming each cavity will be stored in an external file (value of 0) or not (any other value)

2. Algorithm

The implemented algorithm maps an interacting particle system in an equivalent hard-sphere one. The subsequent geometric analysis is based on Delaunay tessellation [14,15]. Once the tessellation is carried out, the accessible volume in each Delaunay tetrahedron is determined. Then the cavities are formed by clustering neighboring Delaunay elements fulfilling a first-passage criterion [11,20]. The Delaunay approach has remarkable analogies with the tetrahedron method developed for the k-space integration in band structure calculations of crystalline solids [21]; for more details see [22].

The suggested algorithm starts with the assignment of an atomic mass and radius to each particle in the system. The atomic radius is given by the sum of the atomic van der Waals radius, r_{vdW}, and the radius of the chosen spherical probe particle, r_P. The r_{vdW} element can be related to the Lennard-Jones σ parameter, i.e. the finite distance where the interatomic potential for dispersion interactions becomes zero; we have $r_{vdW} = 2^{-5/6}\sigma$ [23]. The r_P is user-defined by inserting the value of the "ClusteringRadius" parameter. A complete listing of all user-specified parameters is given in Table 1. Subsequently, the configuration, i.e. the coordinates of the particles of the system at a certain time, is transferred into the simulation box with center at $(0, 0, 0)$. The box is assumed to be three-dimensional and periodic in all spatial directions while its shape is expected to be rectangular, *but not necessarily cubic*.

The Delaunay tessellation is performed by employing the qhull library [24] (http://www.qhull.org). This library is well-known in the field of computational geometry and is adopted as a standard tool for Delaunay analysis (more than 2000 citations). Since the current implementation of qhull treats only non-periodic systems, the simulation box is augmented with its 26 first neighboring images prior to the tessellation. The periodicity is restored by suitably processing the obtained tetrahedra. At first, the tetrahedra with all four atoms outside the simulation box are discarded. Then, the elements lying on the boundaries of the simulation box are considered. Every Delaunay element with three atoms lying outside the simulation box is the periodic image of an element having only one atom outside the simulation box. Therefore, the former elements are also discarded while the atoms of the latter elements located outside the simulation box are replaced by their periodic images which reside inside the box. Similarly, for each Delaunay element with two atoms outside the simulation box, its periodic image is found and deleted. Moreover, the two atoms which are outside the box are replaced by their periodic images, located inside the box.

Before proceeding with the calculation of the accessible volume per Delaunay element, the simulation box is divided into small cubic sub-regions. The length of each sub-region is equal to the maximum atomic radius in the system. Then a neighbor list for each sub-region is constructed. For each Delaunay element, the neighboring atoms, whose spheres overlap with the given tetrahedron, are determined by a sphere–triangle intersection test [25].

The definitions of the unoccupied and accessible volume of a Delaunay element follow the ones proposed by Arizzi et al. [11]. The unoccupied volume of a Delaunay element is the volume of a tetrahedron which is not occupied by any atomic sphere. The unoccupied volume fraction is defined as the ratio of the unoccupied volume to the overall volume of the Delaunay element. In a similar fashion, the accessible volume of a spherical probe particle with radius r_P is defined as the volume of the Delaunay element which is not occupied by any r_P augmented atomic sphere. The accessible volume is estimated by employing a Monte Carlo integration scheme. The irregular shape of the accessible volume elements precludes a direct analytical integration. In the Monte Carlo integration scheme employed in the present report, a geometric point lying inside the considered Delaunay tetrahedron is randomly chosen. Then, it is tested whether the point is located within any of the spheres of the four atoms forming the Delaunay element or any atomic sphere in the immediate vicinity of the tetrahedron. If there is no overlap, the insertion is successful. The number of attempted point insertions is controlled by the "maxMC" parameter. The accessible volume fraction is equal to the ratio of the successful insertions to the attempted ones.

The next step is the clustering of tetrahedra to larger entities where each one corresponds to a cavity. The clustering is performed by employing a "first-passage" criterion [11,26]. The physical interpretation is as follows: If the common surface between two neighboring tetrahedra formed by three atoms is large enough so that a particle residing in the interior of the first tetrahedron can pass through the surface migrating to the interior of the second tetrahedron, then these two tetrahedra should be considered as parts of the same cavity and clustered together. The present implementation discriminates four different cases. In the first one, the atomic sphere of one particle is large enough that it encloses the other two particles. Thus, the surface is completely blocked and the two tetrahedra are not connected. In the opposite limit, the atomic spheres of all three particles are so small that there is no overlap among them. Hence, the surface is open and the two tetrahedra are directly connected. Note that the probe size is already taken into account into the atomic radius. Two more cases have to be examined: (i) there are only two intersections among the three spheres, i.e. the atomic sphere of one particle intersects with the spheres of the two other particles and (ii) there is an overlap between every pair of atomic spheres. In case (i), we first define the joining vector between the two centers of the non-overlapping atomic spheres. Then we test whether all points of the joining vector, which do not belong to any of the non-intersecting spheres, are enclosed by the third sphere. If they are not lying inside the third sphere, the tetrahedra are connected and considered to belong to the same cavity. In case (ii), the standard trilateration procedure is employed. If the point of intersection of any two atomic spheres is located inside the third sphere, the surface is blocked and the tetrahedra are disconnected; otherwise they are assigned to the same cavity.

A number of quantities for each cavity can be calculated. One is the cavity volume, v_C, by summing the individually accessible volumes of all tetrahedra belonging to the given cavity: