Computational Materials Science 118 (2016) 245–250

Contents lists available at ScienceDirect

Computational Materials Science

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

Generation of polycrystalline material at the atomic scale

B. Mantisi

Laboratoire de Physique Théorique de la Matière Condensée, Paris Sorbonne Universités UPMC, BP 121, 4 Place Jussieu, 75252 Paris Cedex 05, France

ARTICLE INFO

Article history:
Received 14 January 2016
Received in revised form 26 February 2016
Accepted 1 March 2016

Keywords:
Polycrystalline materials
Classical molecular dynamics
Grain boundary
Aluminium
Olivine

ABSTRACT

Polycrystalline structure plays an important role in the macroscopic properties of a solid material. In this paper we propose a new code to generate a polycrystalline material at the atomic scale. Our polycrystalline systems are based on the random generation from initial germs. From a mechanical point of view, it brings new features as for example ductility of 3D polycrystalline aluminium. By using molecular dynamics simulations, we also show the effect of temperature on a geological material, olivine, one of the most abundant silicate mineral of the Earth upper mantle. The importance of porous media is also considered (e.g. metals).

1. Introduction

The importance of atomic dislocations on the mechanical behaviour of solid materials is well known and the study of polycrystalline materials is important in that respect. Polycrystalline materials are common in nature. For example, in metallurgy, grain boundaries will generate new macroscopic behaviours compare with a mono-crystalline phase. In geophysics, the polycrystallinity of rocks is a rule. Moreover the Earth mantle being submitted to extreme conditions of temperature and pressure, the influence of the grain boundaries on mechanical and transport properties of rocky mantle is then important to know. A fruitful way to approach the pressure and temperature dependence of these materials is to use molecular dynamics (MD) simulations.

MD simulations are widely used in many fields as mechanics, physics, geology and biology. The main ingredient is the force field (FF), which governs the interaction between particles. The validity of the simulation results depends directly on the accuracy of the FF. Furthermore, for polycrystalline materials the initial configuration of the atoms in the simulation cell has to be as realistic as possible. In general, molecular dynamics codes of the literature give the possibility to build only mono-crystalline structures [1]. Some other codes were developed to build more complex crystalline structures made of proteins [2] or polymers [3]. But, concerning polycrystalline material, users face up to a difficult task. The challenge is to built a polycrystalline structure including a part of randomness. An obvious way would be to proceed as in a real experiment: i.e. cooling a liquid slow enough to let appear and grow mono crystalline germs. This method works well for metals simulated by MD [4] for which single symmetries (fcc or bcc) facilitate the growth from a small nucleus [19]. However, the success of the method is potential dependent and does not work necessarily for every kind of materials (e.g. silicates). Another method consists in building crystalline cells with a Voronoi tessellation [5,6], and assigns to each crystalline area a random orientation. The main advantages of this method is that it is independent of the FF, it is scalable to different sizes and applicable to different kind of simulations (MD and finite element methods for example). However, all grains are separated by planes defined a priori, whereas some roughness is expected at grain boundaries. Moreover, no porosity can be introduced in that way to model polycrystalline structures. Moreover, atomistic simulation of polycrystalline materials has shown already nice results [7] and therefore a code especially devoted to build such initial structure would be a good starting point. In this context, we propose to build a polycrystalline material from the growth of mono-crystals which is stopped when such crystals join each other within the simulation box. The code produces at the atomic scale a random configuration representative of a polycrystalline system. Here, randomness means a random distribution of initial germs with random orientation and random growth. Generated polycrystals are an assembly of several perfect monocrystals, i.e. defects are not introduced inside a grain. In Section 2 we present the basic ideas of the method and how the algorithm works. In Section 3 some details of the code are given. In Section 4 a first example presents the effect of polycrystallinity on the plasticity of aluminium. The second example deals with the partial melting of olivine, a silicate mineral of the Earth upper mantle. We conclude and give some perspectives in Section 5.

http://dx.doi.org/10.1016/j.commatsci.2016.03.002
0927-0256/

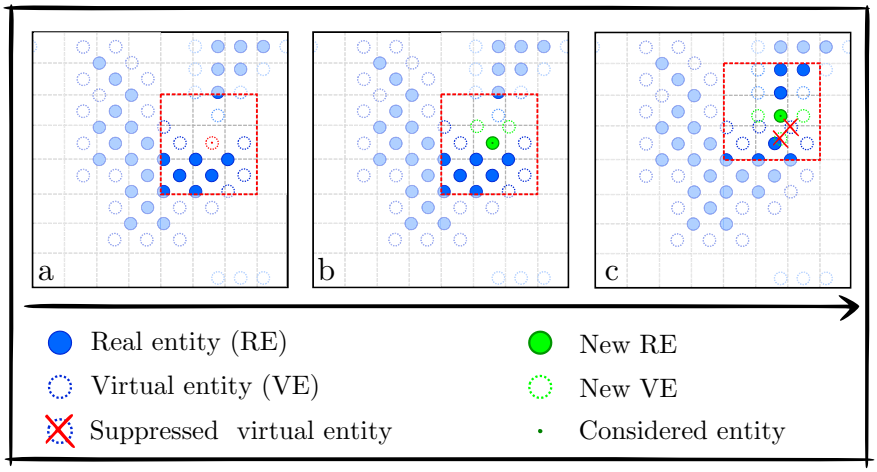

Fig. 1. (a) One VE is selected, (b) and declared as RE. New VE are then defined. (c) Another VE is selected and declared as RE. If it overlap with existing VE, the latter are suppressed. If New VE overlap with existing RE they are not created.

2. Description of the code

The code uses a geometrical criterion based on the unit cell structure (UCS) of the material under investigation. Fig. 1 sketches the process discussed below. The algorithm follows a growth process, which implies to consider two kinds of chemical entities (An entity is the unit cell of the considered crystal or a subunit of this unit cell). Chemical entities that do exist (RE for real entities) and chemical entities that can potentially exist (VE for virtual entities) as first neighbours of REs. The initial state is a distribution of REs (germs) to which we assign a random location and a random orientation in the simulation box. These germs are the first RE in the box. A virtual grid mapped on the crystal symmetry is associated with each germ. The second step is the crystal growth of these initial germs from their associated grid. For that, VE are defined as nearest neighbour of RE on their grid, and then algorithm pick up randomly a VE and declares it as RE. The new RE implies first to delete VEs (i.e VEs from other germs) closer than the minimum distance between entities defined from the UCS and second to declare new VE following the grid of the last defined RE. The process is stopped when the grids are filled taking into account that entities of different grids can not overlap each others. An important task of the code is to deal with the configurations where entities belonging to different grains become close to each other. The growth process is very efficient since all random pick up are accepted, because VE have been defined such that they do not overlap with any other RE. To avoid boundary effects during the simulation, the code considers periodic boundary conditions when writing RE and VE.

Some accelerating features, discussed in the next part, have been implemented, giving then access to large system sizes in an acceptable generating time. The main accelerating feature is a subdivision of the box in sub boxes which decrease the number of computations. It is depicted in Fig. 1 by the red[1] square surrounding a set of sub boxes. Instead of computing all distances and checking a possible overlap with all atoms in the box, the computation deals with a small number of surrounded boxes (more precisely defined in the following). We plot in Fig. 2 the time it takes to generate system sizes reported on *x*-axis for 3-dimension and 2-dimension systems respectively. We note that for a 2 dimensional system, it takes less than one minute on a Intel Xeon E5-2650 @ 2.6 GHz with *gfortran* compiler and just one core, to generate a configuration composed of 200 000 atoms.

[1] For interpretation of colour in Fig. 1, the reader is referred to the web version of this article.

3. Details on the code

3.1. Grain growth

Fig. 3 presents the two dimensional growth of 20 mono atomic germs with a cfc geometry in a 200 × 200 box units (1 box unit = atomic diameter). The separation between grain is imposed to be 2 box units. Since the algorithm is not *a real growth process* because temperature and then diffusion of species are not considered. Concerning the grains statistics evolution, the bigger the grains, the higher the number of VE around have to be considered, i.e more chance they have to be larger than others. By contrast, some mono-crystals are trapped, because they are closely surrounded by other mono-crystals precluding to them any further growth. The density of germs and the space between them when the simulation starts are thus very important.

3.2. Code optimisation

The code has been written to generate many different configurations of several hundred of thousand of atoms. Any loop on all atoms at each building step would be a lost of time. We reduce the computational time using sub boxes paving the simulation cell with sizes of a few angstroms. Then all atoms or entities are labelled in a sub box and the check for non-overlapping is done using nearest neighbour boxes. This is the main feature which renders the code so efficient (A two dimensional box containing around 200 000 atoms is generated in less than one minute). The size of small boxes is related to the nearest neighbour distances within the unit cell of the crystal under investigation. This method applies to any crystal structure.

3.3. Adjustable parameters

The user can easily change the dimension of the box (2 or 3 dimensions), the size of the box, the number of grains, and the space between grains. The space between grains is equal or greater than the typical distance between two atomic species of the crystal unit cell. Whatever is the space imposed between the grains (porosity), the final configuration needs to be minimized in energy with an adequate FF. The code does not provide a minimization method, since existing molecular dynamics codes do it already pretty well. In the forthcoming, we will use LAMMPS for minimizing box generated structures. During the minimization process, the free surfaces minimize their area, porosity is restructured (Fig. 5).